

Acquiring Broad Commonsense Knowledge for Sentiment Analysis Using Human Computation

THÈSE N° 7240 (2016)

PRÉSENTÉE LE 7 OCTOBRE 2016

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE D'INTELLIGENCE ARTIFICIELLE
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Marina BOIA

acceptée sur proposition du jury:

Dr D. Gillet, président du jury
Prof. B. Faltings, Dr P. Pu Faltings, directeurs de thèse
Prof. B. Liu, rapporteur
Prof. Ph. Cudré-Mauroux, rapporteur
Prof. D. Gatica-Perez, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2016

“Knowledge is like a sphere, the greater its volume,
the larger its contact with the unknown”

— Blaise Pascal

To my family...

Acknowledgements

In chronological order, as much as possible... I would like to thank my parents for always loving and supporting me. My brother, for helping me with my first steps in computer science. Aunt Delia, for encouraging me to choose EPFL, for sharing some of her optimism along the way. The EPFL Doctoral School, for awarding me a first year Ph.D. fellowship. Prof. Boi Faltings and Dr. Pearl Pu, for giving me a chance as their Ph.D. student, for precious guidance, feedback, and discussions. But most of all, for invaluable encouragement, patience, and understanding when I needed it most. Alexandra and Stefan, for their pointers with my first Ph.D. milestone - the candidacy exam. Claudiu, for highly appreciated guidance and feedback through most of my Ph.D. years. Bao-Duy, for help with the initial implementation and launches of my games. Prof. Bing Liu, for believing in the potential of my research, for helping me choose a direction to further develop it, for guidance on this new topic. Google, for financially supporting my Adwords experiments and for a great internship experience. Goran, Valentina, and all the LIA and HCI lab members that I have met along the way, for pointers on how to refine my work. Grégoire and Maxime, for their help with translating my games in French. Régis, for the help with translating this thesis's abstract to French. Sylvie, for always being a great help with logistic-related issues, and for always putting a smile on my face when doing so. My defense committee members, for agreeing to review this thesis and for their valuable feedback on how to improve it. Last but not least, my friends. Alina, for being a great flatmate and my partner in travel. Immanuel, for being Big Ngeker. Adrian and Daniel, for accepting me in their long-time friendship circle. Andrew and Valentina, for our lunches, coffee breaks, and nice trips. Régis, for debating tennis matches with me. All, for being my safety net.

Lausanne, 1st September 2016

M. B.

Abstract

While artificial intelligence is successful in many applications that cover specific domains, for many commonsense problems there is still a large gap with human performance. Automated sentiment analysis is a typical example: while there are techniques that reasonably aggregate sentiments from texts in specific domains, such as online reviews of a particular product category, more general models have a poor performance.

We argue that sentiment analysis can be covered more broadly by extending models with commonsense knowledge acquired at scale, using human computation. We study two sentiment analysis problems. We start with document-level sentiment classification, which aims to determine whether a text as a whole expresses a positive or a negative sentiment. We hypothesize that extending classifiers to include the polarities of sentiment words in context can help them scale to broad domains. We also study fine-grained opinion extraction, which aims to pinpoint individual opinions in a text, along with their targets. We hypothesize that extraction models can benefit from broad fine-grained annotations to boost their performance on unfamiliar domains. Selecting sentiment words in context and annotating texts with opinions and targets are tasks that require commonsense knowledge shared by all the speakers of a language. We show how these can be effectively solved through human computation. We illustrate how to define small tasks that can be solved by many independent workers so that results can form a single coherent knowledge base. We also show how to recruit, train, and engage workers, then how to perform effective quality control to obtain sufficiently high-quality knowledge. We show how the resulting knowledge can be effectively integrated into models that scale to broad domains and also perform well in unfamiliar domains.

We engage workers through both enjoyment and payment, by designing our tasks as games played for money. We recruit them on a paid crowdsourcing platform where we can reach out to a large pool of active workers. This is an effective recipe for acquiring sentiment knowledge in English, a language that is known by the vast majority of workers on the platform. To acquire sentiment knowledge for other languages, which have received comparatively little attention, we argue that we need to design tasks that appeal to voluntary workers outside the crowdsourcing platform, based on enjoyment alone. However, recruiting and engaging volunteers has been more of an art than a problem that can be solved systematically. We show that combining online advertisement with games, an approach that has been recently proved to work well for acquiring expert knowledge, gives an effective recipe for luring and engaging volunteers to provide good quality sentiment knowledge for texts in French.

Our solutions could point the way to how to use human computation to broaden the compe-

Abstract

tence of artificial intelligence systems in other domains as well.

Key words: commonsense knowledge acquisition, human computation, crowdsourcing, gamification, games with a purpose, sentiment analysis, sentiment classification, fine-grained opinion extraction

Résumé

Malgré le succès de l'intelligence artificielle dans de nombreuses applications propres à certains domaines, il y a toujours un écart important avec l'intelligence humaine dans beaucoup de problèmes demandant du bon sens. L'analyse automatique de sentiments en est un exemple typique : bien qu'il existe des techniques qui permettent de raisonnablement rassembler les sentiments de textes pris d'un domaine spécifique, les modèles généraux ont des performances plutôt limitées.

L'analyse de sentiments peut être traitée de façon plus vaste en élargissant les modèles existants avec des connaissances de sens commun acquises à grande échelle à l'aide du calcul humain. On étudie deux problèmes d'analyse de sentiments. Nous commençons avec la classification de sentiments au niveau des documents, dont le but est de déterminer si un texte exprime dans l'ensemble un sentiment positif ou négatif. En élargissant des classificateurs avec les polarités des mots de sentiments dans le contexte correspondant, on peut les amener à l'échelle de domaines plus généraux. Nous étudions aussi la fouille d'opinions à granularité fine, qui essaie d'identifier des opinions individuelles dans un texte, avec leurs cibles. Les modèles d'extraction peuvent être améliorés par l'acquisition d'annotations à granularité fine pour un vaste domaine, ce qui peut ensuite mener à de meilleures performances quand appliquées à des domaines nouveaux. Sélectionner des mots de sentiments dans le contexte et annoter des textes avec les opinions et leurs cibles sont des tâches qui nécessitent du bon sens. Nous montrons comment ces tâches peuvent être résolues à l'aide du calcul humain. Nous illustrons comment définir des petites tâches qui peuvent être complétées par de nombreux travailleurs, puis assemblées en une base cohérente de connaissances. Nous montrons aussi comment recruter, former, et captiver des travailleurs, puis comment vérifier efficacement leur travail pour obtenir des connaissances de qualité élevée. Nous montrons comment ces connaissances peuvent être efficacement intégrées dans des modèles qui peuvent s'appliquer à des domaines généraux et également avoir de bonnes performances dans des domaines nouveaux.

Nos tâches sont conçues en tant que jeux avec possibilité de gagner de l'argent en récompense, ce qui permet de garder les travailleurs impliqués dans l'activité. Nous recrutons les travailleurs sur une plateforme payante de crowdsourcing, où nous pouvons atteindre un grand nombre de travailleurs actifs. C'est une recette efficace pour acquérir des connaissances en anglais, une langue connue par la majorité des travailleurs sur la plateforme. Pour acquérir des connaissances sur les sentiments dans d'autres langues nous soutenons qu'il y a un besoin de concevoir des tâches suffisamment attrayantes pour des travailleurs volontaires externes à

Résumé

la plateforme de crowdsourcing, en se basant uniquement sur le plaisir d'accomplir la tâche. Cependant, recruter et captiver des volontaires s'est révélé être plus un art qu'une science. Nous montrons que la combinaison de publicité en ligne avec des jeux, une approche qui a été récemment démontrée comme fonctionnant bien pour acquérir des connaissances expertes, donne une recette efficace pour obtenir des connaissances de bonne qualité pour des textes en français.

Nos solutions pourraient montrer la voie pour utiliser le calcul humain pour élargir les compétences des systèmes d'intelligence artificielle à d'autres domaines.

Mots clefs : acquisition de connaissances, calcul humain, crowdsourcing, gamification, games with a purpose, analyse de sentiments, classification de sentiments, fouille d'opinion à granularité fine

Contents

Acknowledgements	i
Abstract (English/Français)	iii
List of figures	xi
List of tables	xiii
1 Introduction	1
2 Background	7
2.1 Sentiment Analysis	7
2.1.1 Document-level Sentiment Classification	8
2.1.2 Fine-grained Opinion Extraction	16
2.2 Human Computation	23
2.2.1 Overlap with Crowdsourcing	23
2.2.2 Worker Recruitment and Motivation	24
2.2.3 Task Understanding	26
2.2.4 Quality Assurance	26
2.3 Sentiment Knowledge Acquisition with Human Computation	27
2.3.1 Text Corpora	27
2.3.2 Sentiment Lexicons	28
2.3.3 Fine-grained Annotations	29
2.3.4 Sentiment Knowledge Acquisition with Volunteers	30
3 Knowledge Acquisition for Scalable Sentiment Classification	31
3.1 Introduction	31
3.2 Document-level Sentiment Classification	34
3.2.1 Bigrams Context	34
3.2.2 Human-generated Context	35
3.3 Human Computation Task	38
3.3.1 Task Structure	38
3.3.2 Worker Motivation	39
3.3.3 Tutorial	40
3.3.4 Quality Assurance	40

Contents

3.3.5	Worker Recruitment	46
3.4	Empirical Results	46
3.4.1	Dataset	46
3.4.2	Task Setup	48
3.4.3	Context Statistics	50
3.4.4	Context and the Lexicon-based Method	51
3.4.5	Context and the Supervised Learning Method	52
3.5	Conclusions	55
4	Knowledge Acquisition for Generalizable Opinion Extraction	63
4.1	Introduction	63
4.2	Fine-grained Opinion Extraction	65
4.2.1	Opinion and Target Pairs	66
4.2.2	Targets Alone	68
4.3	Human Computation Task	69
4.3.1	Task Structure	69
4.3.2	Quality Assurance	70
4.4	Empirical Results	73
4.4.1	Dataset	73
4.4.2	Task Setup	73
4.4.3	Annotation Evaluation	74
4.4.4	Model Evaluation	76
4.5	Conclusions	80
5	Sentiment Knowledge Acquisition with Volunteers	83
5.1	Introduction	83
5.2	Recruiting and Engaging Volunteers	86
5.2.1	Task Design Exploration	86
5.2.2	Game Metaphor Comparison	93
5.2.3	Worker Recruitment	94
5.3	Empirical Results	95
5.3.1	Dataset	95
5.3.2	Exploratory Phase	95
5.3.3	Game Metaphor Comparison	102
5.4	Conclusions	108
6	Conclusions	109
6.1	Summary	109
6.2	Limitations	110
6.2.1	In Context Acquisition and Integration	110
6.2.2	In Fine-grained Annotation Acquisition and Integration	111
6.2.3	In Sentiment Knowledge Acquisition with Volunteers	111
6.3	Future Work	112

Contents

Bibliography	126
Curriculum Vitae	127

List of Figures

2.1	Background. Example of a digital camera review from Amazon.com	8
2.2	Background. Example of a hotel review from TripAdvisor.com	8
2.3	Background. Example of a vacuum cleaner review from Amazon.com	9
2.4	Background. Example of a digital camera review containing pros and cons summaries from Epinions.com	21
3.1	Context acquisition. Main game interface	38
3.2	Context acquisition. Tutorial quiz that asks workers to identify a phrase with a negative polarity	40
3.3	Context acquisition. Tutorial quiz that asks workers to identify a context which makes a phrase positive	41
3.4	Context acquisition. Tutorial explanation of the game rounds, scoring, and puzzles	42
3.5	Context acquisition. A simple example of how the score updates vary in time as workers keep submitting the answer (<i>good, , positive</i>)	44
3.6	Context acquisition. A more complex example of how the score updates vary in time as workers mostly submit the answer (<i>small, room, negative</i>) with some exceptions stating the reverse polarity	44
3.7	Context acquisition. Hierarchical structure of the training set	48
3.8	Context acquisition. Distribution of workers in terms of the number of solved rounds (for the hotels game launch)	49
3.9	Context acquisition. Summary of the game quality survey	51
3.10	Context acquisition. Error of the lexicon with the sentiment model extension. The Hu and Liu lexicon alone, then extended with the separate human-generated context (hgc) at level 3, and finally with the combined hgc at level 1	52
3.11	Context acquisition. Error of supervised models trained at the five hierarchy levels, from level 5 (individual categories) to level 1 (products and hotels combined)	53
3.12	Context acquisition. Error of supervised models with the feature space extension. First, word models extended with hgc. Then, extension of word and bigram models	55
4.1	Annotation acquisition. Main game interface	71
4.2	Annotation acquisition. Example of a training round in the sandbox stage . . .	71
4.3	Annotation acquisition. Average annotation f-scores	76

List of Figures

4.4	Target extraction f-scores for models trained with syntax and full features, in the individual domains (id), union of domains (ud), and cross-domain (cd) setups	78
5.1	Volunteer participation. Animal puzzles game in the design exploration phase	87
5.2	Volunteer participation. Village explorer game in the design exploration phase	89
5.3	Volunteer participation. Labyrinth explorer game in the design exploration phase	90
5.4	Volunteer participation. Animal puzzles game in the metaphor comparison phase	91
5.5	Volunteer participation. Village explorer game in the metaphor comparison phase	92
5.6	Volunteer participation. Labyrinth explorer game in the metaphor comparison phase	93
5.7	Volunteer participation. Examples of advertisements for the games ran in the exploratory phase	96
5.8	Volunteer participation. Sentiment classification performance in the exploratory phase	101
5.9	Volunteer participation. Advertisement text used in the metaphor comparison phase	101
5.10	Volunteer participation. Sentiment classification performance in the metaphor comparison phase	106

List of Tables

2.1	Background. Sample words from the Hu and Liu [40] sentiment lexicon	9
2.2	Background. Sample word weights identified by a Support Vector Machine trained on digital camera reviews	13
2.3	Background. Sample opinions and targets extracted from hotel reviews	17
3.1	Context acquisition. Sizes of the training and test sets for each category	47
3.2	Context acquisition. Details of the game launches	48
3.3	Context acquisition. Sample phrase and context pairs obtained with human computation	57
3.4	Context acquisition. Error of the lexicon with the sentiment model extension. Left, Hu and Liu [40] lexicon alone. Middle, improvement with the separate hgc at level 3. Right, improvement with the combined hgc at level 1	58
3.5	Context acquisition. Error of the lexicon with the sentiment model extension. Left, Hu and Liu lexicon extended with the individual words in the context at level 3 (hgc-iw), then with all the context at level 3 (hgc-all). Right, extended with hgc at level 1	58
3.6	Context acquisition. Error of the supervised method with the sentiment score extension. Left, individual words model at level 1. Right, extended with hgc at level 1	59
3.7	Context acquisition. Error of the supervised method with the sentiment score extension. Left, individual words model at level 1 extended with hgc-iw at level 1. Right, extended with hgc-all at level 1	59
3.8	Context acquisition. Error of supervised models with the feature space extension. Left, individual words model at level 3, then improvement with hgc at level 3. Right, models at level 1	60
3.9	Context acquisition. Error of the supervised method with the feature space extension. Left, individual words model at level 3 extended with hgc-iw at level 3, then with hgc-all at level 3. Right, models at level 1	60
3.10	Context acquisition. Error of the corpus method with the feature space extension. Left, words and bigrams model at level 3, then improvement with hgc at level 3. Right, models at level 1	61
4.1	Annotation acquisition. Features of the Pair SVM for opinion and target extraction	66

List of Tables

4.2	Annotation acquisition. Example of a game round and individual annotations obtained, along with the final aggregate annotations (in bold)	73
4.3	Annotation acquisition. Number of sentences and opinion target pairs in the gold annotations and in workers' aggregate annotations	75
4.4	Annotation acquisition. First, annotation f-scores based on agreement on the opinion and target components, respectively. Next, joint agreement on both the opinion and the target (opn-trg). Finally, full agreement on all annotations components (full)	75
4.5	Annotation acquisition. Opinion and target extraction f-scores of Double Propagation and the three variants of the Pair SVM model: individual domains (id), union of domains (ud), and cross-domain (cd)	76
4.6	Annotation acquisition. Target extraction f-scores for models trained on the individual domains	79
4.7	Annotation acquisition. Target extraction f-scores for models trained on the union of domains	79
4.8	Annotation acquisition. Target extraction f-scores for models tested across domains	79
4.9	Annotation acquisition. Sample annotations obtained for camera, mattress, and restaurant reviews	81
5.1	Volunteer participation. Statistics of the advertisement campaigns we ran in the exploratory phase (in 2015)	97
5.2	Volunteer participation. Overlap between the three lexicons generated during the exploratory phase and the reference lexicon	100
5.3	Volunteer participation. Statistics of the first advertisement campaign run in the metaphor comparison phase	103
5.4	Volunteer participation. Statistics of the second advertisement campaign run in the metaphor comparison phase	104
5.5	Volunteer participation. Statistics of the lexicons generated during the metaphor comparison phase	105
5.6	Volunteer participation. Overlap between the lexicons generated during the metaphor comparison phase and the reference lexicon	105

1 Introduction

Why Commonsense Knowledge?

As Cambria et al. [16] point out, human intelligence is “the human ability to harness common-sense knowledge gleaned from a lifetime of learning and experience to make informed decisions. This allows humans to adapt easily to novel situations”. Knowledge acquisition is thus central to bridging the gap between human and artificial intelligence (AI), and this has always been the most important challenge for AI. So far, successes have been obtained for specific domains, both through knowledge engineering, as in expert systems, and through machine learning, as in speech recognition for a single speaker. However, acquiring knowledge that is valid over a broad domain of application has remained elusive. As Cambria et al. further remark, in novel situations, AI “fails catastrophically due to a lack of situation-specific rules and generalization capabilities”. Personal assistants such as Siri [5] cannot handle general conversations, while speech recognition for the general population still has very high error rates. As an example of a problem that would benefit from more broadly applicable common-sense knowledge, we consider automated sentiment analysis: the problem of aggregating the sentiments (opinions) expressed in a text.

Why Sentiment Analysis and Why Human Computation?

Sentiment analysis has many practical applications. Internet users frequently post their thoughts on blogs, Facebook, Twitter, or other social media platforms. They also write reviews in which they share their experiences with products and services. Properly aggregating the sentiments expressed in these texts would offer priceless insight into what people think: manufacturers could better understand how to improve their products to meet their clients’ needs; hotel managers could learn how to better their services; doctors could figure out where to adapt their patient skills; politicians could better understand their electorate and what is expected from them. The general population would also profit, by improved understanding of the options available to them in various life circumstances: what brand to choose for a particular product? what DVD to rent on a movie night? what family doctor to decide on?

Chapter 1. Introduction

what candidate to vote for in an election? Sentiment analysis can even offer insights into what people expect about the future: who will win a particular election [87]? what actors and movies will win at the Oscars [123]? how will stock prices fluctuate [12]? Knowing what to anticipate might spoil the fun when one is reading a book or watching a movie. In general, though, it is a useful advantage to have when one is trying to choose an optimal strategy, which is why we have election polls, betting rates, or meteo forecasts.

There is thus great value in effectively aggregating sentiments from texts, and this has prompted researchers to develop automated solutions. Sentiments have been mined at different granularity levels: at the document or sentence-level [90], or at the finer-grained level of expressions and even individual words [115]. Existing techniques perform reasonably well on texts from specific domains, such as online reviews of a particular product category, but drop in performance when they need to handle a broader domain. Nevertheless, humans can use their common sense to easily identify sentiments in texts regardless of their topic. We thus argue that sentiment analysis can be covered more broadly by extending models with commonsense knowledge acquired at scale, using human computation (see Chapter 2 for an introduction to sentiment analysis and human computation, as well as an overview of existing attempts to acquire sentiment knowledge with human computation).

Main Contribution

In this thesis, we target two sentiment analysis problems: document-level sentiment classification and fine-grained opinion extraction. The former problem aims to establish whether a text as a whole expresses a positive or a negative sentiment. Fine-grained opinion extraction focuses on extracting individual opinion expressions from a text, along with their corresponding targets. On the one hand, sentiment classifiers could benefit from knowledge about the contexts that impact the orientation (polarity) of the sentiments expressed by particular words. This would allow them to scale and effectively aggregate text sentiments in a broader domain. On the other hand, opinion extraction models could benefit from fine-grained annotations for texts in a broad domain. If properly exploited, these annotations would allow to identify patterns for opinion and target extraction that are more effective on unfamiliar domains. As our overall contribution, we show: how such knowledge can be effectively obtained using human computation; and how it can be integrated into sentiment analysis approaches to expand their coverage.

Scalable Sentiment Classification with Human-generated Context We first focus on the sentiment classification problem. Classifiers for sentiment fall into two categories. In supervised methods, a classifier is trained using machine learning on a corpus of text. To keep the learning complexity manageable, features are generally limited to the most frequent words in the training corpus. Such classifiers can obtain good performance as long as their application domain remains relatively small. The other line of work is lexicon-based. These approaches rely on sentiment lexicons - lists that summarize the sentiment words most common in a

language, along with their positive or negative polarities. Here, a text is classified by matching the sentiment words it contains. An overall label is then inferred based on the proportion of the two word categories. These approaches can be applied broadly, but their accuracy is much lower than that of supervised methods and generally insufficient for practical tasks.

A key to improving scalability lies in how these methods model sentiment knowledge. Both methods consider words individually. This does not work so well on texts containing sentiment words whose polarity is ambiguous outside their context. For example, the word *cold* does not express a concrete sentiment on its own. However, in contexts like *beer* or *pizza*, it gains a positive and negative polarity, respectively [31]. Sentiment lexicons typically do not refine the polarities of words with contexts. This is why they consistently perform poorly on broad domains. Conversely, supervised methods can automatically learn polarity scores for individual words. On a narrow domain, words appear in only a few contexts, so these methods perform well by learning context-dependent polarities. For instance, a machine learning algorithm separately trained on *pizza* and *beer* reviews might pick up the word *cold* as positive and negative, respectively. However, this is no longer the case on broader domains, where these methods also harm performance. When trained on reviews of pubs, generally expected to serve both *pizza* and *beer*, the algorithm will not know whether to consider *cold* as positive or as negative. This general model will thus perform worse than its specialized counterparts.

Therefore, sentiment classifiers should incorporate context by including longer word combinations. However, this makes the feature space increase substantially and, while these longer features could be learned from data, one would need a huge corpus. This is why, so far, even attempts restricted to learning the polarities of word pairs have reported mixed results. To help lexicon and supervised approaches scale, we need to find a way to reliably acquire context. Here, we acknowledge that, unlike machines, humans can use their common sense to correctly select both sentiment words and their disambiguating contexts, even from very short sentences. For instance, given the text: *I had cold pizza and warm beer for dinner, how sad*, human can easily spot that *cold* is negative in the context of *pizza* and that *warm* is also negative in the context of *beer*. As a first contribution, we show: how such knowledge can be effectively acquired using human computation; and how it can enhance sentiment classifiers such that these scale to a broad domain. This line of work is presented in detail in Chapter 3 and has been in part published in [9, 10, 11]:

[9] Boia, M., Musat, C. C., and Faltings, B. Acquiring commonsense knowledge for sentiment analysis through human computation. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web* (2014), pp. 225–226

[10] Boia, M., Musat, C. C., and Faltings, B. Acquiring commonsense knowledge for sentiment analysis through human computation. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence* (2014), pp. 901–907.

[11] Boia, M., Musat, C. C., and Faltings, B. Constructing context-aware sentiment lexicons with an asynchronous game with a purpose. In *Proceedings of the 15th International*

Conference on Computational Linguistics and Intelligent Text Processing (2014), pp. 32–44.

Generalizable Opinion Extraction with Human-generated Annotations We next focus on the fine-grained opinion extraction problem. Here, existing approaches fall under two categories. Unsupervised methods are used to generate lexicons of frequent opinion expressions, as well as lexicons with the targets of opinions (in product reviews, these would be parts or properties of the reviewed items). These methods usually apply hand-crafted syntactic rules. For instance, they rely on the observation that opinions are typically adjectives and that their targets are usually nouns (e.g. *wonderful flavor*). Unsupervised approaches work well within their training domain, but only when they are applied to a small corpus. On the other hand, supervised methods are used to pinpoint the exact location of opinion expressions in a text, as well as that of their targets. Many approaches are only partially supervised: they use opinion and target lexicons to highlight candidate passages in the text, and only then apply supervision to find which opinions correspond to which targets, based on the syntactic properties of the text. Some approaches are fully supervised, bypassing the need for lexicons in candidate extraction. Similar to the unsupervised approaches, supervised methods only have a high accuracy within their training domain.

A key to improving generalization lies in the scope of the training corpus. Many opinion and target words are specific to particular domains and do not transfer to others. For example, the words *taste* or *aroma* might be the target of opinion in beer reviews but they will most probably not appear in reviews of electronic products. By broadening the training corpus, extraction models should become familiar with more varied opinion and target features, thus increasing their efficiency on new domains. However, because they are based on hand-crafted heuristics, unsupervised methods pick up many false positives when applied to large text collections. Moreover, even if the training corpus is broad, the generated lexicons are still likely to have an incomplete coverage on a new domain. Therefore, these methods harm both precision and recall. On the other hand, supervised methods require a training corpus of sentences meticulously annotated with the opinions and targets they contain. So far, such detailed annotations have been expensive to obtain, with only a handful of participants needing to go over thousands of texts. Even if these were available at scale, supervised approaches would still generalize poorly. The partially supervised methods rely on opinion and target lexicons that are often generated with unsupervised approaches. They are thus likely to similarly harm performance. In particular, these methods harm recall by remaining within the bounds of these lexicons, even though, on domains with reduced lexicon coverage, syntactic cues could by themselves restrict the number of false negatives. Conversely, the fully supervised methods tend to primarily involve word features in the decision making process, even if syntax cues are also included. Therefore, on new domains, these methods also generate many false negatives, similar to their partially supervised counterparts.

Given the inherent limitations of unsupervised approaches, we can more realistically hope to improve generalization through fully supervised methods. Here, a first hurdle is obtaining

fine-grained annotations at scale. A second hurdle is how to better integrate syntax features, such that these take the lead when word features do not have coverage on new domains. We acknowledge that, similar to the context acquisition task, humans can rely on their common sense to correctly pinpoint opinion and target passages in texts. For example, given the sentence *This beer has a wonderful fruity taste*, humans can easily highlight *wonderful fruity* as the opinion expressed about the target *taste*. As a second contribution, we show: how fine-grained annotations can be efficiently acquired at scale using human computation; and how supervised methods can better exploit the syntactic patterns revealed by these annotations to improve generalization. We describe this line of work in detail in Chapter 4 (note that, at the time of writing, this work was not published).

Sentiment Knowledge Acquisition with Volunteers To tackle the context and fine-grained label acquisition problems, we define small tasks that can be independently solved by many human workers so that their answers can be aggregated into a single coherent knowledge base. We show how to lure these workers to our tasks, how to engage them to participate, and how to quickly train them so that they understand what is required from them. We also show how to effectively perform quality control to obtain sufficiently high-quality knowledge.

In particular, we rely on a paid crowdsourcing platform, where we can reach out to a large pool of active workers. Moreover, we engage them not only with payments but also through enjoyment, by designing our tasks as games played for money. This combination gives an effective recipe for acquiring sentiment knowledge in English, a language that is known to the vast majority of workers on this platform. However, given the demographics of crowdsourcing platforms, other languages are less accessible to paid workers. This hints that, for the moment, our solution would be less effective when acquiring knowledge in other languages, which in sentiment analysis have received comparatively little attention. We thus argue that we also need to design tasks that reach out to workers outside the paid crowdsourcing platform.

The problem is that recruiting and engaging volunteers has so far been more of an art than a problem that can be systematically approached to achieve good results. There have been several success stories, most notable of which are Wikipedia or Duolingo, platforms that engage Internet users to maintain a free online encyclopedia or translate the Web. Such tasks can gain momentum through word of mouth or through exposure in the media. Moreover, they appeal to workers by touching on their altruistic side or by hooking them with game elements. Nevertheless, these examples are more of an exception rather than the rule. More recently though, it has been shown that volunteer workers can be effectively recruited by advertising tasks online, on sites such as Google Search. It seems that this is an effective method to reach out to workers likely to participate in tasks for expert knowledge acquisition. Moreover, it appears that extending the advertised tasks with game elements engages workers to provide such knowledge with high accuracy. We thus inquire whether this can also be an effective solution for acquiring commonsense knowledge. To underline the advantage of relying on volunteers, we aim to acquire sentiment knowledge in French. As a third contribution, we show

Chapter 1. Introduction

that combining online advertisements with games that engage workers through puzzles or exploration metaphors is an efficient recipe for acquiring such knowledge with good accuracy. This line of work is presented in Chapter 5 (note that, at the time of writing, this work was not published).

Our solutions could point the way to how to use human computation to broaden the competence of artificial intelligence systems in other domains as well.

2 Background

The goal of this thesis is to investigate how human computation can be employed to acquire commonsense knowledge for sentiment analysis. We start by introducing the sentiment analysis problem, focusing on the two sub-problems that we aim to tackle: document-level sentiment classification and fine-grained opinion extraction. We then present the human computation paradigm along with the main concerns involved in designing tasks that effectively harness the power of workers. Finally, we describe how human computation has so far been used to acquire knowledge for sentiment analysis.

2.1 Sentiment Analysis

The sentiment analysis problem (also referred to as opinion mining) aims to automatically interpret and summarize the sentiments (also called opinions) expressed in user-generated texts, such as online reviews of products or services or social media posts. A substantial part of the sentiment analysis research has been conducted on online reviews, and these are the types of text we deal with in this thesis as well.

Bing Liu [69] gives a thorough formalism of the sentiment analysis problem, which we exemplify here on review texts. The author defines an entity as the product or service with respect to which sentiments are expressed, such as the digital camera described by the review in Figure 2.1 or the hotel reviewed by the text in Figure 2.2. An entity is comprised of several aspects, which can be its parts or its properties. For example, aspects of a digital camera include: its zoom and its viewfinder as parts, as well as its image quality or size as properties. Bing Liu then moves on to explain that a sentiment is an expression like *sharp* or *extremely clear*, which appear in the review in Figure 2.1. A sentiment can target either the entity itself or one of its aspects. For example, the previous sentiment expressions are about the picture quality. Moreover, a sentiment has a polarity that can be positive (expressing a favorable attitude) or negative (expressing an unfavorable attitude). For instance, the two sentiment expressions above are positive. Furthermore, a sentiment pertains to its holder (the person expressing it, typically the author of the review) and is conveyed at a particular point in time.

Chapter 2. Background

★★★★★ Great little camera

May 28, 2014

This camera exceeded all of my expectations. It is small, lightweight and very easy to handle with one hand. The pictures are sharp, extremely clear. It has a built in macro lens for shooting small objects/flowers/insects. Excellent camera.

Figure 2.1: Background. Example of a digital camera review from Amazon.com

“Perfect chalet break”

○○○○○ Reviewed March 24, 2016

Stayed for four nights in this cosy but spacious chalet. Had a fantastic time skiing and catching up with friends. The chalet hosts were fantastic - full of useful ski tips, put on the most delicious spreads every night, took us everywhere we needed to go in the minibus, and pulled out all the stops to help when we had a...

[More](#) ▾

Figure 2.2: Background. Example of a hotel review from TripAdvisor.com

Within this formalism, the sentiment analysis problem aims to summarize texts in terms of the sentiments it contains, and all their defining attributes: target, polarity, holder, and timestamp.

Depending on how the sentiments expressed in a text are aggregated, we can distinguish several sentiment analysis sub-problems. Opinions can be mined at different granularity levels: at the document or sentence level, or at the finer-grained level of individual opinion expressions. In this thesis, we are concerned with two sub-problems: document-level sentiment classification and extraction of individual opinion expressions along with their targets. In what follows, we review these two sub-problems and their existing approaches. Our exposition is partially based on the work of Bing Liu [69], to which we refer the interested reader for a more detailed literature review.

2.1.1 Document-level Sentiment Classification

At the document level, the goal is to automatically infer the polarity of a piece of text: whether the sentiments expressed in the text are overall positive or negative. For example, for the text shown in Figure 2.3, one would aim to infer an overall positive polarity. This task is typically studied on reviews, which are texts that tend to discuss the advantages and disadvantages of a single entity - the item being reviewed. This problem becomes more challenging or may even be ill-posed on other types of texts, such as blog posts, whose less restricted format does not guarantee that these express sentiments regarding a single entity.

The polarity of a review is typically derived by reasoning about the polarity expressed through the sentiment words that it contains. More specifically, if the text contains words that are predominantly positive, we can infer that, on the whole, it expresses a positive sentiment,



November 7, 2015

Powerful suction, a little **trouble** getting under beds and furniture due to its size but still a **good** value.

Figure 2.3: Background. Example of a vacuum cleaner review from Amazon.com

word	polarity
adequate	+1
blissful	+1
clean	+1
erroneous	-1
adverse	-1
conceited	-1

Table 2.1: Background. Sample words from the Hu and Liu [40] sentiment lexicon

and vice versa. Therefore, it is common for sentiment classification approaches to rely on knowledge about positive and negative sentiment words. For example, if we know that the words *good* and *powerful* are positive and that *trouble* is negative, we can derive that the vacuum cleaner review in Figure 2.3 has an overall positive polarity. Depending on where the knowledge about the polarities of sentiment words comes from, we can distinguish lexicon-based and supervised learning approaches.

Lexicon-based Approaches

Lexicon-based methods use existing sentiment lexicons, which enumerate sentiment words along with their positive or negative polarities. There are several sentiment lexicons available that one can readily use, such as the General Inquirer lexicon [110], the OpinionFinder lexicon [130], or the lexicon of Hu and Liu [40] (Table 2.1 shows some sentiment words sampled from the Hu and Liu lexicon). If such a lexicon is available, then one can classify a document by summing up the polarities of the sentiment words it contains. If the result is greater than zero, the text can be labeled with a positive polarity, otherwise it can be considered negative.

Sentiment lexicons can be manually compiled by a handful of annotators, who can be either experts in the field or specially trained to understand the task. For example, OpinionFinder has been, at least partially, created by annotating the words from a predefined vocabulary with their polarities. Kim and Hovy [54] relied on three annotators to label a set of adjectives and adverbs. Similarly, Hatzivassiloglou and Mckeown [34] labeled adjectives.

Sentiment lexicons can also be created with automatic methods. One option is to exploit the knowledge captured in resources such as traditional dictionaries or WordNet [82] (further described below). For example, one can use synonymy and antonymy relations between words or the dictionary definitions of terms. An approach is to start from a few sentiment words whose polarities are known (like *good* and *bad*) and to iteratively expand the set of

known sentiment words by propagating their polarities to their synonyms and antonyms. The intuition is that words that are synonyms with positive terms are also likely to have a positive polarity. Similarly, terms that are synonyms with negative sentiment words are likely to be negative themselves. On the other hand, words that are linked through antonymy relations are likely to have opposing polarities. For instance, Hu and Liu [40] used this heuristic to compute the polarities of the adjectives occurring in a corpus of electronics reviews. This choice was based on the intuition that adjectives are very likely to express sentiments. Similarly, Kim and Hovy [54] applied this heuristic to a seed list containing both adjectives and verbs. In addition, they extended this approach with a method that assessed the polarity strength of newly discovered words. Blair-Goldensohn et al. [8] proposed a similar method. Another possibility is to additionally exploit term dictionary definitions. For instance, Adreevskaia et al. [1] started by bootstrapping sentiment words through synonymy and antonymy relations. They then extracted additional sentiment words whose definitions included sentiment terms that had been picked up during the previous step.

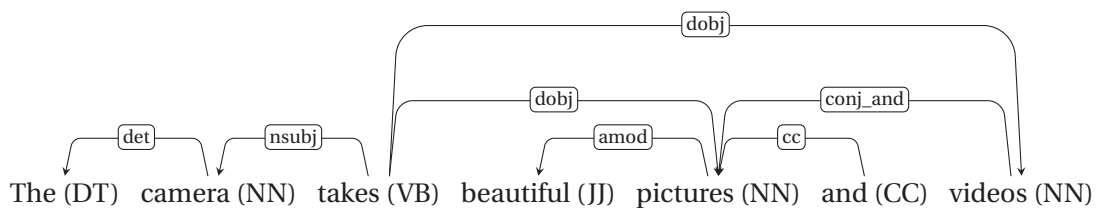
An alternative is to learn sentiment lexicons from data. Some approaches use the intuition that positive sentiment words are more strongly associated with other words that are positive than with words that are negative, and vice versa. For instance, Turney [115] computed the polarities of words as the difference between the strength of their semantic association with the positive word *excellent* and with the negative word *poor*. He estimated the semantic association of two words as their Pointwise Mutual Information [22], which was computed based on the terms' frequencies in the Web pages returned by a search engine. Other approaches generate sentiment lexicons by exploiting syntactic patterns in text corpora (further detailed on below). For example, a possible heuristic is that sentiment words that are linked by conjunctions (e.g. *and*) tend to have the same polarity, whereas words that are linked by disjunctions (e.g. *but*) tend to have opposite polarities. This idea was verified by Hatzivassiloglou and McKeown [34]. Based on this, the authors proposed a model that learned to distinguish whether two adjectives had the same or opposite polarities depending on whether they were linked by a conjunction or a disjunction. They then described a method that clustered words with similar polarities. The algorithm generated two clusters such that, as much as possible, words that were predicted to have the same polarity were placed in the same cluster. Therefore, positive sentiment words were segregated from negative ones.

Popescu and Etzioni [93] combined these alternatives. They applied Turney's method to generate an initial set of sentiment words and used this set to assess how positive, negative, and neutral words were distributed in their data. Based on this, they assigned an initial probability distribution for the polarity of each word. They then iteratively refined these polarity distributions. In every iteration, the distribution of each word was updated such that it was consistent with those of its neighboring terms (established based on synonymy and antonymy relations or conjunctions and disjunctions consistency rules). The authors applied this method to compute the polarities of words that were likely to convey sentiments, which they identified through their syntactic relations with a set of known aspect terms.

WordNet and Syntactic Patterns WordNet is a lexical database for the English language. It organizes words in synonym sets (synsets) - groups of semantically related words. Each synset comes with a definition (gloss), listing the possible meanings of the words it contains. Moreover, synsets are linked through various semantic relations. These links can mark part-of, is-a, or antonymy relations between synsets. Because WordNet captures word definitions as well as their synonymy and antonymy relations, it has often been employed in sentiment lexicon generation.

Syntax studies the principles that dictate sentence construction in a language [129, 88]. The rules indicating what sentences are correct are normally enumerated in a grammar. The most basic elements of a grammar are called parts of speech - categories of words that play a similar role in the construction of sentences [128, 88]. For example, in English, nouns tend to be subjects or objects of verbs, adjectives tend to describe nouns, and adverbs tend to modify verbs. Based on part of speech classes, a grammar could for example specify how the subject, verb, and object follow one another in a sentence [129].

The structure of a sentence can be automatically determined using a natural language parser [109]. A parser can identify the words that are the subjects or the objects of verbs, which adjectives modify nouns, which adverbs modify verbs, or which words are connected through prepositions and conjunctions. Therefore, in the process, a syntactic parser also identifies the parts of speech of the words in a sentence. A commonly used tool is the Stanford Parser [109, 55, 76]. When presented with a sentence like *The camera takes beautiful pictures and videos*, this parser may output the tree below:



The tree nodes are the words in the sentence, tagged with their part of speech. The links are called syntactic dependencies. A dependency links a dependent word (also called modifier) to the term it modifies - the governor (also called head). These dependencies are typed [75]. For instance: *nominal subject(nsubj)* dependencies link a subject to its predicate, the subject typically being a noun or pronoun and the predicate either a verb or adjective (preceded by a copula); *adjectival modifier (amod)* dependencies link an adjective to a noun; *adverbial modifier (advmod)* dependencies connect an adverb to its verb; *direct object (dobj)* dependencies link an object to its verb; *negation (neg)* dependencies link a negation adverb to the word it modifies, often an adjective; *conj_and* marks a link between two words created by an *and*.

We have seen lexicon generation approaches that rely on syntactic patterns, which can be matched in sentences with the help of a parser. Many other sentiment analysis approaches do so as well. It is common to reason about word part of speech tags and how these interact in a

Chapter 2. Background

sentence through adjacency or through the dependencies that link them. Section 2.1.2 gives more details about syntax heuristics that have been used to extract opinions and their targets.

Supervised Learning Approaches

Another approach is to apply supervised machine learning algorithms, such as Naive Bayes, Maximum Entropy, or Support Vector Machines (SVMs) [23]. If a corpus of texts annotated with positive or negative polarities is available, these algorithms can be used to learn a sentiment classifier that knows how to label new, unseen documents. When presented to the learning algorithm, a text is typically represented as a bag of features: a vector marking the presence or frequency of various features within the document. Very often, these features are words from a predefined vocabulary, such as the most frequent words in the training corpus or the terms from an existing sentiment lexicon. Learning algorithms will typically assign a polarity weight to each feature, based on their distribution in the positive and negative training texts. When presented with an unseen document, the learned classifier will reason about the weights of the features it contains to derive an overall polarity. For example, when enhanced with a linear kernel, a Support Vector Machine learning on digital camera reviews could identify feature weights similar to those shown in Table 2.2. To classify an unseen document, the model will weight the presence or frequency of each feature, sum up these values, then threshold the result at zero. Pang et al. [90] were the first to treat document-level sentiment classification as a supervised learning problem, and the approach has since been widely used [125, 133, 50, 68].

Text Corpora for Sentiment Classification

To train supervised methods and to evaluate all sentiment classification approaches, one needs a corpus of texts annotated with their overall polarity. A lot of corpora for document-level sentiment classification consist of online reviews. As mentioned previously, these texts have the advantage of expressing sentiments about a single entity. Moreover, reviews typically come with a title, a main body of text, and a star (numerical) rating, in the range of one to five or one to ten. This rating indicates the author's overall attitude towards the reviewed item. Therefore, another advantage of using reviews is that their star rating can be used as a gold standard. It is also possible to manually label documents using a few annotators. However, this is typically done for shorter texts, such as individual sentences [54, 8, 29].

Need for Better Scalability

Sentiment classification approaches do not scale well to broad domains. This is because both lexicon and supervised learning methods tend to only reason about the polarities of individual words. However, some texts are more challenging to classify, given that they contain sentiment words that are ambiguous by themselves and require context for their polarity to be correctly

word	weight
pleased	0.52
glad	0.41
excellent	0.36
atrocious	-0.63
disappointment	-0.69
returning	-0.77

Table 2.2: Background. Sample word weights identified by a Support Vector Machine trained on digital camera reviews

interpreted. For instance, we cannot say anything about the polarity of the word *long*, but in the context of *battery life* we can interpret it as positive, whereas in the context of *focus time* it becomes negative [25]. Another issue is that even unambiguous sentiment words, like *good*, can have their polarities flipped by neighboring terms. For example, *not good* or *hardly any good* convey a negative polarity. In broad domains, sentiment words can appear in multiple contexts, which is why sentiment classifiers that rely on individual words do not scale well.

Context in Lexicon-based Approaches In general, existing sentiment lexicons do not detail the polarities of sentiment words in specific contexts. Because sentiment words can have multiple relevant contexts, the required level of details makes it tedious to acquire contextual knowledge from a handful of annotators (as an exception, Popescu and Etzioni [93] used two annotators to label combinations consisting of either adjectives or adverbs along with aspects; Lu et al. [72] also described a similar annotation process). Moreover, the above lexicon generation approaches typically derive the polarities of individual words (one exception is the work of Turney [115], who proposed a raw model context, by learning the polarity of word pairs matching a few syntactic patterns, such as combinations of adjectives and nouns or of adverbs and verbs). Because most sentiment lexicons do not explicitly model context, these methods typically have a low performance.

A few dictionary and data methods that generate context-independent lexicons can also be used to obtain context-dependent ones. For example, Popescu and Etzioni [93] mined online reviews to automatically identify the polarities of sentiment words in the context of aspects. As previously described, they first used an algorithm that iteratively updated the context-independent polarities of sentiment words such that they were consistent with those of its neighboring terms. They then further refined these polarities by using a similar iterative algorithm, this time applied to pairs consisting of sentiment words and aspects.

Ding et al. [25] approached the same task. They used the intuition that people write reviews in a coherent way. This means that: the sentiment words within a sentence are likely to have the same polarity (unless a disjunction is used); consecutive sentences are also likely to convey sentiments with the same polarity. Therefore, the authors learned the polarities of sentiment words in the context of aspects by relying on their vicinity with unambiguous sentiment words

Chapter 2. Background

whose polarities were already known. They then used synonymy and antonymy relations to propagate the learned context-dependent polarities to new sentiment words.

Lu et al. [72] also learned the polarities of sentiment words in the context of aspects. To construct a lexicon, the authors proposed several heuristics. First of all, they employed the polarities of sentiment words from context-independent lexicons. Secondly, the authors used the intuition that the polarities of sentiment words appearing in a review are indicative of the star rating attached to that review. Thirdly, they also relied on sentiment consistency heuristics involving conjunctions, disjunctions, and negations. Moreover, they exploited synonymy and antonymy relations between words. They used these four signals to define a set of constraints over the polarities of sentiment words in context. The authors learned the polarities that best satisfied these constraints.

Makki et al. [73] produced a similar lexicon. They described an iterative approach that started from a few seed sentiment words with known polarities. The authors considered: nouns and noun phrases to be aspects; words modifying aspects through specific syntactic patterns to be sentiment words. In each iteration, new sentiment words and aspect pairs were identified based on the known ones. Once a pair was extracted, its polarity was computed based on known sentiment words that modified the same aspect in the same review. Here, the main intuition was that, within the same document, a reviewer will consistently refer to an aspect through sentiment words that carry identical polarities.

Wu and Wen [131] also learned context-dependent polarities. The authors focused only on a very limited set of sentiment words that are typically used as quantifiers, such as *big* or *small*. The goal was to identify nouns in the context of which such words gain a polarity that is positive or negative. Their intuition was that, for some nouns like *salary*, people have a positive expectation. That is, they expect the entity referred to by the noun to appear in a large quantity. Conversely, some nouns like *price* have a negative expectation, and people expect the referenced entity to appear in small amounts. Therefore, when such nouns are combined with quantifier words, the resulting phrase conveys a positive or negative polarity: *big salary* versus *small salary*. The authors proposed several word patterns that people normally use in order to express such expectations. They instantiated these patterns with several nouns and used a search engine to estimate the saliency of the resulting phrases. Based on this, they were able to learn the expectation of nouns, which in turn allowed them to compute the context-dependent polarities of the quantifier words. Other methods for context-dependent lexicon generation were proposed by Fahrni and Klenner [26] and by Bross and Ehrig [14].

In another attempt to model context, some approaches used supervised learning to predict the polarities of sentiment words in the context of a longer phrase or a sentence. For example, Wilson et al. [130] remarked that the prior polarities of sentiment words (as indicated by traditional sentiment lexicons) may change due to the influence of neighboring terms within a sentence. The authors described an annotation scheme in which subjective expressions appearing in sentences were labeled with their contextual polarity (positive, negative, or

neutral). They then proposed a supervised learning approach for predicting the polarities of subjective expressions in context. They achieved this by combining two classifiers: a first one distinguishing whether an expression was neutral or conveyed a polarity; a second model aiming to predict the polarity of an expression. The second model used various features such as the subjective expression to be classified, its prior polarity according to a sentiment lexicon, whether the expression was negated, or whether the expression was connected through specific syntactic relations to other sentiment expressions in the sentence. Similarly, Choi and Cardie [20] tried to predict the polarity of longer sentiment expressions by using compositional semantics to model how individual words within a sequence interacted with each other to give an overall polarity. On a similar note, Socher et al. [107] modeled sentiment compositionality at the sentence level. They proposed a recursive neural network that learned: how to represent words as weight vectors; and how the weight vectors corresponding to the words in a sentence could be gradually combined to predict the polarities of longer phrases, eventually outputting the polarity of the whole sentence. It is also possible to derive contextual polarities using hand-crafted heuristics. For example, Kennedy and Inkpen [50] employed a set of rules that adjusted the original polarity and strength of sentiment words when these were in the vicinity of contextual valence shifters [91] from a predefined list. These are special terms like negations, intensifiers, and diminishers, which can inverse, strengthen, or alleviate the polarity of a sentiment word.

Context in Supervised Learning Approaches Supervised learning approaches use individual words as features. In very narrow domains, the lexicons that are learned from data can identify words that have domain-specific polarities. For example, in reviews about compact digital cameras, the word *small* might only appear in contexts that give it a positive polarity, since small compact cameras are desirable when traveling. Therefore, a supervised learning algorithm might assign a positive weight to this feature and will correctly use it to classify unseen reviews. However, this approximation of context will not be sufficient on larger domains, where sentiment words can have both positive and negative polarities, depending on their context. For instance, in a domain that also includes vacuum cleaner reviews, the word *small* might be negative in the context of canisters, given that this is not a desirable property for vacuums. As a result, a model learning on this larger domain will not know whether this feature is positive or negative, thus harming performance. This is why supervised learning approaches have a good performance on narrow domains, but degrade as their application domain broadens.

In an attempt to incorporate a raw model context, some supervised learning approaches also experiment with other features such as longer word combinations, or incorporate negations or other types of syntactic relations between words. Pang et al. [90] tried to combine individual words (unigrams) with pairs of consecutive words appearing in the corpus (bigrams). However, they remarked that adding longer word combinations did not have a positive effect on performance. On the other hand, Wang and Manning [125] also studied bigrams and proved the opposite. Xia and Zong [133] also analyzed the effect of longer word combinations. More

specifically, they defined features based on bigrams or syntactic relations between words. To improve performance, they generalized these word pair features such that the first term was replaced with its part of speech tag. They remarked that simply combining unigrams with these generalized features did not help performance. Instead they proposed to have separate classifiers for these feature sets and to combine their output using an ensemble method. They found this approach to work better than simple feature combination.

Kennedy and Inkpen [50] also tried to model context by complementing unigrams with longer word combinations. However, they focused on those bigrams that contained a regular word along with contextual valence shifters. The authors remarked that adding these special bigrams had a small but statistically significant impact on performance. Li et al. [67] also experimented with contextual valence shifters. However, rather than spotting them with predefined keywords or other heuristics, they trained a model to identify sentences that contained contextual valence shifters. They then created two datasets, one based on texts with contextual valence shifters and one not. They trained separate models on these datasets and combined their output to produce a final classification.

Human-generated Context for Better Scalability We have seen approaches that generated context-dependent lexicons or extended supervised learning methods with features that captured context to some extent. However, reliably learning contextual knowledge from data is hard. Modeling context means extending the feature space from individual words to at least word pairs. This means that the number of features increases substantially. To make things more manageable, previous approaches limited the feature space based on syntactic patterns or based on adjacency. However, this means that valuable word combinations were probably missed. Moreover, even when the feature space is restricted, it is still hard to learn reliable polarities from data, as one would still need a very large annotated corpus. And indeed, as we have seen, bigrams have so far led to mixed results. On the other hand, humans can use their common sense to easily identify sentiment words and their contexts. Therefore, we use human computation to acquire contextual knowledge that helps sentiment classification methods scale to broad domains.

2.1.2 Fine-grained Opinion Extraction

At the finer-grained level, the goal is to identify all the individual opinion expressions that appear in texts, along with their corresponding targets. As it was previously mentioned, the target of an opinion can be an entity or an aspect of an entity. Similar to document-level sentiment classification, this problem has been mostly studied on reviews, where entities are the reviewed items, and aspects are its parts or properties. Given a text collection, one facet of this problem is to construct two lexicons: one containing the most salient opinion expressions appearing in the corpus, and another containing the most frequent opinion targets. For example, given some hotel reviews, we could extract the opinions and targets shown in Table 2.3. A second facet of the problem aims to pinpoint the occurrence of each

opinion	target
loved	kitchen
spacious	elevator
accommodating	shower
renovated	bathroom
upgraded	bedroom
smelled	breakfast
stained	lunch

Table 2.3: Background. Sample opinions and targets extracted from hotel reviews

opinion expression in the corpus and to find the target that each opinion refers to. For instance, given this sentence from a hotel review: *The check-in took forever and the staff was not helpful*, we would ideally locate one opinion *took forever* about the target *check-in*, and a second one *was not helpful* about the target *staff*.

There are two types of approaches to this problem. Some are unsupervised and tend to solve the first facet of the problem - generating lexicons with opinions and targets. These methods also help with solving the second facet - pinpointing opinion expressions in texts as well as their corresponding targets. This is because opinions and targets can be matched in texts using lexicons and then paired using hand-crafted syntax or proximity heuristics. Other approaches are supervised, and tend to solve the second facet of the problem. Some methods are only partially supervised and pair opinions and targets matched with lexicons, while others are fully supervised and bypass the need for dictionaries.

Unsupervised Learning Approaches

Unsupervised approaches usually rely on syntax heuristics. A first intuition is that some of the most frequent adjectives, such as *good* or *excellent*, are likely to be opinions. Moreover, the nouns and noun phrases most often mentioned in reviews, such as *zoom* or *battery life*, are likely to be targets. A second intuition is that opinions are typically used to refer to targets, and thus opinions and targets appear in each other's vicinity. For example, Hu and Liu [40] extracted the most frequent targets from product reviews. They considered a set of noun or noun phrases occurring together in a review sentence to be an itemset. They applied association rule mining [2] to this itemset and considered the resulting frequent itemsets to be targets. The authors also extracted infrequent aspects. As mentioned in Section 2.1.1, they generated an opinion lexicon by selecting the adjectives appearing in sentences that contained frequent targets. Using this opinion lexicon, they identified infrequent targets as the nouns and noun phrases found in the vicinity of opinions, restricting to those sentences that did not already contain a frequent aspect.

Popescu and Etzioni [93] studied the same problem. They improved on the approach of Hu and Liu by making sure not to extract nouns and noun phrases that were not actually

aspects. For a candidate aspect, the authors computed the strength of its association with the considered product class. They used word patterns that captured the part-whole relation between an aspect and the product class. For instance, for the scanner class, they used patterns like *[aspect] of scanner*, *scanner comes with [aspect]*, *scanner has [aspect]*. They instantiated these patterns with candidate aspects, then assessed how plausible they were by running Web queries. As mentioned in Section 2.1.1, the authors also generated a lexicon of opinions by iteratively refining the polarities of words participating in certain syntactic relations with known aspects.

Blair-Goldensohn et al. [8] also improved the idea of Hu and Liu. They extracted nouns and noun phrases only from sentences that had been labeled as positive or negative by a sentence-level sentiment classifier, or from sentences that contained candidate targets that participated in syntactic patterns indicative of the presence of an opinion, such as a noun following an adjective. As previously mentioned, the authors also generated a lexicon of opinions by exploiting word synonymy and antonymy relations.

Rather than separately computing opinion and target lexicons, Wang and Wang [124] exploited the idea that these can be jointly extracted in an iterative process. Their algorithm started from a small set of opinion seeds. At each iteration, the method first identified targets based on the known opinions. The authors considered nouns and noun phrases as candidate targets and extracted those that often co-occurred with the set of known opinion words. An iteration then proceeded to extract opinions from known targets. Adjectives were considered to be candidate opinions, and the authors extracted those that frequently co-occurred with the known targets.

Qiu et al. [94] described a similar iterative approach, called Double Propagation. However, instead of exploiting opinion and target adjacency, the authors used a parser and defined several syntactic dependency patterns. More specifically, they defined patterns for the extraction of word pairs that would include an opinion and a target, two opinions, and two targets. These patterns referred to dependency types that linked nouns to adjectives, adjectives to adjectives, and nouns to nouns, respectively. Using these patterns, the method started from a small set of seed opinion words and iteratively expanded the sets of known opinions and targets. In each iteration, four steps were performed: extracting new opinions using the known opinions and the defined dependency patterns, extracting new targets using the known opinions, extracting new opinions using the known targets, and extracting new targets using the known targets. Zhang et al. [137] extended the Double Propagation approach with additional patterns meant to increase the recall of target extraction. In addition, they introduced a post-processing step that ranked target candidates by their importance and frequency.

On a different note, rather than relying on adjacency or syntax heuristics, Liu et al. [70] mined alignments between opinions and targets with a word-based translation model in a monolingual setup. The authors also proposed an additional step in which they ranked candidate targets according to a confidence score. In [71], this model was extended by guiding the alignment of words connected through certain dependency types.

Supervised Learning Approaches

Most supervised methods start by pinpointing candidate opinions and targets in review sentences. Supervision is then used to establish which candidate opinions correspond to which candidate targets. Zhuang et al. [139] focused on extracting opinion and target pairs from movie review sentences. As targets of opinions, they considered movie aspects, which they defined as either movie elements (e.g. screenplay, music) or movie-related people (e.g. actor, director). They identified candidate opinions and targets using lexicons. The opinion lexicon was compiled from word statistics in their annotated corpus, then enlarged using synonymy relations. The target lexicon contained a few terms grouped into movie element classes and was enriched with the names of movie-related people. To decide whether candidate opinions and targets were connected, the authors mined the frequent chains of syntactic dependency types and part of speech tags that linked the opinion and targets in their annotated corpus.

Kobayashi et al. [56] also identified candidate opinion and targets using lexicons. They generated these lexicons from a corpus of reviews, using the semi-automatic method proposed in [58]. For a candidate opinion, the task was to then decide which of the candidate targets corresponded to it. The authors used a tournament model [41] that compared two candidate targets in reference to the considered opinion. The problem was modeled as a binary classification, where one candidate target or the other can prevail (a Support Vector Machine with a second-order polynomial kernel was used). Multiple comparisons were conducted, until there was only one candidate target remaining. To represent a candidate target in reference to an opinion, the model used features such as part of speech tags, the number of words between the two, whether they were connected by a syntactic relation, or whether they appeared together in a predefined co-occurrence list.

Kobayashi et al. [57] started with the identification of candidate opinions. These were matched using an opinion lexicon, which was generated from a review corpus, also using the semi-automatic method in [58]. As candidate targets, the authors considered all the non-opinion words in a sentence. Given a candidate opinion, the best target was found using syntax and co-occurrence patterns that were learned from data.

To identify candidate opinions, Wu et al. [132] used the OpinionFinder lexicon. To identify candidate targets, the authors used a lexicon that they compiled by extracting all the noun and verb phrases in their corpus. Each phrase in this list was scored using a review language model and dropped if this score was under a predefined threshold. To decide which candidate opinions were linked to which targets, the authors implemented a phrase dependency parser. Phrase dependency parsing aims to segment a sentence into phrases, such as verb or noun phrases, then to link them with directed arcs. For a candidate opinion and target pair, the sub-tree rooted at their lowest common ancestor in the phrase parse tree was considered for classification as positive or negative. To achieve this, the authors defined a new tree kernel over phrase dependency trees, which they incorporated within an SVM that classified trees as positive or negative.

Chapter 2. Background

Kessler and Nicolov [51] considered the opinion and target expressions as already annotated and available. They defined their task as that of establishing which of multiple candidate target annotations belonged to a particular opinion. They used a Ranking SVM [49] that learned to sort the target annotations with respect to a particular opinion annotation. For a reference opinion and a candidate target, the model used features such as the number of tokens between the opinion and the potential target, these tokens along with their part of speech tags, the dependency types on the shortest path linking the opinion and target, or the part of speech tags of the opinion and target.

Other methods also incorporate supervised learning when pinpointing the candidate opinions and targets. This is typically done by classifying the words in a sentence as opinions, targets, or something else. Here, a popular choice are Conditional Random Fields (CRFs) [61], which can learn how to label words based on their context. For instance, a linear-chain CRF considers words in a sequence and predicts a label for a token based on the labels of adjacent words. Yang and Cardie [134] incorporated this approach in their work. They chose to jointly learn how to extract candidate opinions and targets and how match opinions to their targets. Their approach had several components. One component was a Conditional Random Field that extracted opinions and targets. Another component was a Logistic Regression model [80] that decided whether a pair containing an extracted opinion and target belonged together. This model used features such as the two words involved and their part of speech tags, the number of tokens that separated them, the path in the dependency tree that linked them, the strength of subjectivity according to the OpinionFinder lexicon, and so on. The third component was a set of constraints that linked the output of the two models and allowed them to be optimized together. Choi et al. [19] described a similar approach.

On the same note, Jin and Ho [47] used a lexicalized Hidden Markov Model (HMM) [99] to label tokens in a sentence as opinions, targets, or something else. However, to match the extracted opinions to their targets, they replaced supervision with proximity heuristics. The approach of Li et al. [66] was similar. Instead of HMMs, the authors extracted opinions and targets using CRFs. Apart from the standard linear-chain CRF, which considered words in a sequence, they proposed three more variants. They described a Skip-Chain CRF, which incorporated information about the conjunctions and disjunctions that connected opinion or target words within a sentence. The model had two types of edges: linear edges as in the linear-chain model; and skip edges in between words connected by conjunctions or disjunctions. They also proposed a Tree CRF model, in which nodes corresponded to words in the sentence dependency tree and edges linked the nodes that were connected through a direct dependency. Finally, they also described a Skip-Tree CRF model, in which they combined the tree and skip edges. As Jin and Ho, to match the extracted opinions to their corresponding targets, they relied on proximity heuristics.

Choi and Cardie [21] focused only on extracting opinion expressions, along with their polarity and intensity attributes. The authors proposed a model which, given the words in a sentence, output a sequence of labels that were the conjunction of a polarity value (positive, neutral,

★★★★☆ February, 15 2014

Pros: Excellent still image quality, Good autofocus and shutter lag performance

Cons: No external battery charger, Cost

Figure 2.4: Background. Example of a digital camera review containing pros and cons summaries from Epinions.com

negative) and an intensity value (high, medium, low). To achieve this, their model combined a hierarchical parameter sharing technique [15, 138] with a Conditional Random Field.

Jakob and Gurevych [45] focused only on extracting the targets of opinion expressions and considered the latter as already annotated and available. They used a linear-chain CRF. This model was evaluated in single domain and cross-domain settings and shown to outperform the approach in [139]. Similarly, in the SemEval 2014 aspect extraction sub-task [92], the top performing teams proposed Conditional Random Fields.

Yu et al. [135] also extracted only targets. However, rather than using a model that sequentially labeled words, the authors classified terms individually. They extracted nouns appearing in reviews that contained pros and cons sections (such a review is shown in Figure 2.4) as high-precision targets. They used this as a training set for a One-Class SVM [74] that learned to discriminate nouns that were targets from those that were not. They then applied this model to extract targets from the main body of texts of reviews.

Text Corpora for Fine-grained Opinion Extraction

To train supervised methods and to evaluate all opinion extraction approaches, one needs a fine-grained annotated corpus. Most approaches obtain it using a handful of annotators. Wiebe et al. [127] described a detailed annotation scheme for sentences extracted from news articles. The authors relied on the notion of private states [97], which they refer to as “internal states that cannot be directly observed by others”. For example, these include opinions, beliefs, thoughts, or emotions. Their annotation scheme captured the components of private states that were explicitly expressed in the text: an attitude along with its experiencer, its target, and its properties (e.g. intensity, polarity). The annotation scheme also captured the components of private states that were indirectly conveyed in the text: the text span implying the attitude, its source, as well as its properties. The authors presented the results of an agreement study with three annotators, who were trained by reading a detailed instruction manual, practicing annotation on a few documents using pencil and paper, then by learning how to use the annotation tool.

Toprak et al. [113] described the annotation of sentences coming from reviews of online universities and online services. The annotation scheme distinguished between: explicit expressions of opinions and polar facts (facts that can be objectively verified but still imply an opinion towards something). For the former, they annotated the opinion expression span

Chapter 2. Background

in the text, its target and holder, and labeled it with its polarity and strength. For polar facts, they annotated its target and labeled it with a polarity. The authors presented the results of an inter-annotator agreement study with two annotators.

Several similarly annotated datasets exist. Hu and Liu [40] annotated sentences from reviews of electronics. The sentences were labeled with the targets of opinions, along with their corresponding polarities and strengths (it seems that the authors labeled the corpus themselves). Wachsmuth et al. [122] employed two expert annotators to mark hotel reviews with all the occurrences of hotel aspects. Ganu et al. [27] annotated sentences from restaurant reviews in a similar fashion. However, the sentences were annotated with coarse aspect categories (e.g. food, price, service) and not with the actual aspect mentions in the text (they used three annotators). Blair-Goldensohn et al. [8] proceeded similarly. For the SemEval 2014 competition [92], a corpus was created with sentences from restaurant and laptop reviews. These sentences were labeled with the aspects of the reviewed entities and with the polarity expressed towards these (two annotators were used). Zhuang et al. [139] annotated sentences from movie reviews with pairs of opinion expressions and their targets (four annotators were used). Kessler et al. [52] created a similar dataset, with sentences from blog posts about digital cameras and cars. The annotation scheme was more detailed, in that it also specified relations between targets, like part-of or instance-of. Moreover, the modifiers of opinion expressions were also marked, including neutralizers, negations, or intensifiers (four annotators were used). Other corpora with opinion and target annotations are described in [124, 56, 57, 70].

Need for Better Generalization

Approaches to fine-grained opinion extraction are effective on their training domain but do not perform well on new domains. This is because many target words and even some opinion words are domain-specific and do not transfer to new domains. For example, targets like *check-in*, *staff*, *bedroom*, and *bathroom* are frequent in hotel reviews. So are opinions like *fast*, *friendly*, and *central*. While these expressions will be highly relevant for hotel reviews, they will be of no use in a lot of other domains. Because unsupervised methods extract the most frequent words that comply with certain syntax heuristics, they are likely to construct opinion and target lexicons that are of little use in new domains. On the other hand, a lot of the supervised methods we have seen rely on such lexicons to pinpoint candidate opinions and targets. Supervision is used only when deciding which opinions correspond to which targets, based on syntactic cues. Because they rely on lexicons, these methods are also likely to harm performance on new domains. Finally, even supervised methods that do not employ lexicons tend to generalize poorly. For example, a Conditional Random Field that uses both syntax and word features tends to assign most of the weight to word features. It will thus not generalize to new domains when these words do not transfer.

One way to improve generalization is by training on a broader corpus with data from multiple domains. However, because they are based on hand-crafted heuristics, unsupervised methods tend to pick up a lot of errors when applied to big corpora [137]. Moreover, even if the

training corpus is broad, the resulting lexicons are still likely to have only partial coverage on new domains. On the other hand, supervised methods require training data with fine-grained annotations, but obtaining these has so far been expensive, with only a handful of annotators having to cover thousands of sentences. Even when annotations are available for a broader domain, some supervised models can still generalize poorly. The partially supervised approaches typically rely on candidate lexicons generated with unsupervised methods. They thus suffer from similar problems. In particular, these methods harm recall by not leveraging syntactic cues alone in cases where opinion and target lexicons have no coverage. Moreover, fully supervised methods are still susceptible to memorizing word features as opposed to also relying on syntactic cues. Therefore, these models will still perform poorly on new domains.

Human-generated Annotations for Better Generalization We can more realistically hope to improve generalization using supervised methods. A first key is obtaining a broadly annotated corpus. Here, we argue that humans can use their common sense to easily annotate texts with opinions and targets. Therefore, we use human computation to acquire fine-grained annotations for multiple domains. A second key is using such fine-grained annotations to train a model in a way that does not harm performance on unfamiliar domains. We thus describe a supervised model that, unlike a CRF, can leverage these annotations to learn both syntax and word features that do not harm performance on new domains.

2.2 Human Computation

2.2.1 Overlap with Crowdsourcing

Law et al. [64] define human computation as a paradigm that involves “using human effort to perform tasks that computers cannot yet perform, usually in an enjoyable manner”. This idea was introduced by von Ahn in his Ph.D. dissertation [117]. According to Quinn and Bederson [96], the terms *human computation* and *crowdsourcing* are often used interchangeably, but the latter is a different paradigm. Howe was the first to use the term crowdsourcing, in a Wired magazine article [38]. He later defined this paradigm as “the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call” [37].

While they are two different paradigms, human computation and crowdsourcing do have some overlap, when a task can be performed by both machines and humans [96], such as generating a sentiment lexicon. An automatic sentiment lexicon generation method will not be fully accurate in creating such a resource, whereas humans are much better at identifying words that carry sentiment. Therefore, we may want to design a human computation task to create this lexicon with the help of many workers. Conversely, generating such a lexicon by relying on a few experts will be time consuming, so we might be better off crowdsourcing this task to many workers. Therefore, a task engaging workers in the creation of a sentiment lexicon

can be considered both a human computation and a crowdsourcing task. In general, tasks that require commonsense knowledge acquisition for sentiment analysis or natural language processing lie at the overlap between human computation and crowdsourcing.

2.2.2 Worker Recruitment and Motivation

There are several main concerns in how to effectively design human computation tasks. One concern is how to recruit workers and how to motivate them to participate once they land on the task page. An option is to financially motivate workers. In this case, it is common to post tasks on paid crowdsourcing platforms, such as Amazon Mechanical Turk¹ or CrowdFlower². On these platforms, businesses or individuals can create small tasks that are solved by a large population of workers, in exchange for financial rewards (the Amazon Mechanical Turk platform is described in more detail below).

The alternative is to motivate users to voluntarily participate. One way to engage voluntary workers is to appeal to their altruistic side and to make them aware that their participation is valuable for the society. For instance, by contributing to platforms like Wikipedia³, workers can help create an invaluable resource: an online encyclopedia that is free and available to anyone with an Internet connection. Another successful example is the Zooniverse platform⁴, where workers can contribute to research in a wide spectrum of fields, such as astronomy (e.g. by labeling galaxies in pictures) or zoology (e.g. by annotating penguins in images). It is also possible to turn tasks into a learning experience. For example, on Duolingo⁵, workers learn a new language and at the same time help translate the Web.

Another option is to make tasks enjoyable, typically by designing them as games that engage workers in a way that elicits the desired output. One of the first successful examples was the ESP game [118], in which workers played in teams of two and had to agree on phrases that best described an image shown to them. The output of this game was a large set of annotated images. Another successful game is Foldit⁶, where workers need to find the optimal three-dimensional structure of proteins. These are chains of amino acids that do not form a straight line but fold in a particular way, which dictates the proteins' functions. Knowing the optimal structure of a protein could help scientists better understand proteins involved in diseases. Workers playing this game were able to find solutions that outperformed automatically generated ones [77]. Several other human computation games have been designed, targeting image tagging [121, 119, 35], video or music annotation [105, 65], or commonsense knowledge acquisition [120, 17, 16]. In general, enhancing tasks with game elements has been shown to have a positive impact on worker engagement [84].

¹<https://www.mturk.com>

²<https://www.crowdfunder.com>

³<https://www.wikipedia.org>

⁴<https://www.zooniverse.org>

⁵<https://www.duolingo.com>

⁶<https://fold.it>

Relying on paid workers guarantees that a task will be completed in a reasonable amount of time. Crowdsourcing platforms have large pools of active workers looking to complement their income by solving such tasks. This means that, if the financial incentive is sufficiently appealing with respect to the task complexity, there will be no problem in recruiting enough workers to complete the task in a short time. On the other hand, it is not clear how a community of volunteers can be recruited. In some cases, this happens organically, through word of mouth. Other solutions would be to recruit workers through viral marketing on social media [6], or to rely on exposure in traditional media. However, there is a certain amount of randomness involved in whether or not a task relying on volunteers gains momentum, and there is no clear recipe that guarantees this. To fix this issue, Ipeirotis and Gabrilovich [43] proposed a middle-ground solution: attracting voluntary workers through paid online advertising. The authors showed that volunteers can be attracted by running ads through the Google Adwords platform⁷ (described in more detail below). They quantified the quality of the work done by each worker and sent this feedback to the advertising platform, allowing it to optimize ad placement. The authors also explored which gamification elements could be used to more effectively engage workers. They concluded that, for tasks that require expert knowledge, volunteers could perform high quality work. Moreover, the cost for running online ads was below what one would spend on platforms such as Amazon Mechanical Turk. This approach has been also employed by Kobren et al. [59]. However, they showed that paid workers were, on the contrary, able to provide better expert knowledge than the volunteers recruited with online ads.

Amazon Mechanical Turk and Google Adwords

On the Amazon Mechanical Turk platform, there are two types of users. On the one hand, there are the requesters - businesses and individuals that create and post small tasks, typically referred to as HITs (Human Intelligence Tasks). A HIT is a set of questions, such as a text to be annotated or an image to be labeled, that can be answered in exchange for a small financial reward. On the other hand, there is a large pool of workers - users that solve these HITs in exchange for the promised payments.

Requesters typically create groups (batches) that consist of several similar HITs. For instance, several texts of the same kind, needing annotation. As a result, all the HITs in a batch will have to advertise the same financial reward. A batch needs to have a short title and description, as well as a few keywords that summarize what the HITs are about. In addition, requesters need to specify the number of workers that should complete each HIT (note that a worker can complete several HITs from the same batch). Finally, requesters can decide which workers can have access to their task, for example by filtering them based on the country.

Workers typically start by browsing the list of HIT batches accessible to them. Based on the description and payment, a worker can decide whether she wants to approach (accept) a

⁷<https://adwords.google.com>

Chapter 2. Background

HIT in a particular group. If a worker completes a HIT, the requester needs to review the quality of her work and decide whether or not to approve that submission. Workers receive the advertised payment only in case of an approval. In addition, when workers do not want to complete a HIT that they have accepted, they have the option to return it, in which case that HIT becomes available to other workers. Finally, each worker has a record that keeps track of various statistics, such as how many HITs they have submitted or the percentage of approved submissions. Based on this, requesters also have the option to make HITs accessible only to workers with a good reputation.

The Google Adwords platform can be used to run online advertisement campaigns. An advertiser configures a campaign by creating several advertisements along with a set of corresponding keywords. An ad typically consists of a short title and text description. In addition, it can be clickable, in which case it leads to a page that the advertiser wants to promote. Ads are shown when an Internet user looks up one of the chosen keywords on the Google Search page. Ads can also be shown when a user browses a page from the Google Display Network (partner sites that display Google ads on their pages). More precisely, an ad will be presented to the user if the page's content is relevant to the chosen set of keywords. An advertiser can configure a campaign with further parameters. For example, she can restrict a campaign to certain countries and provinces. Finally, she can configure a daily budget. Here it is important to note that, if ads are clickable, the advertiser is charged only when users interact with them.

2.2.3 Task Understanding

Another concern is making sure that workers understand the task. This can be achieved by creating a tutorial that explains the basic rules and concepts [112]. The tutorial can be a separate entity that precedes the task. Another option is to embed it in the task interface, in the form of short instructions that are attached to the main interface controls, guiding workers on how to proceed.

2.2.4 Quality Assurance

Yet another concern is controlling the quality of answers. This can be done before workers even have access to the task. For example, on Amazon Mechanical Turk, workers have the reputation scores that indicates their overall performance. Therefore, if workers are recruited on such platforms, one option is to allow only the ones that have a good track record to access the task [96]. Because they can ensure that workers understand the task, tutorials also have an impact on quality. Moreover, tutorials can include interactive quizzes that workers need to solve correctly in order to be given access to the real task [112].

During the task, quality can be controlled through certain game elements, such as intelligent scoring mechanisms that reflect the quality of answers. Here, we remind the human computation games that group workers in teams of two, require teammates to independently answer

2.3. Sentiment Knowledge Acquisition with Human Computation

the same question, and reward them with points only if they agree on their answers [118]. A twist of this strategy is to have workers individually solve tasks and reward them if their answers agree with those of previous workers. Such schemes incentivize workers to solve the task according to their best effort and generally have a positive effect on quality [84].

Quality can also be controlled after the task is completed. For instance, a task can interleave regular questions with gold-standard questions for which the correct answers are known in advance. After workers complete the task, their performance can be assessed on these gold questions, and the answers given by bad workers can be dropped [96]. In addition, the answers that multiple workers have indicated for the same question can be aggregated in order to reach a higher quality [106].

2.3 Sentiment Knowledge Acquisition with Human Computation

Sentiment analysis relies on resources acquired through either traditional tasks, involving a small number of participants, or through human computation tasks, engaging many workers in answering commonsense questions. These resources consist of text corpora and words annotated with their polarities and of texts annotated at the finer-grained level of individual opinion expressions and their targets. Traditional approaches have been mentioned in Sections 2.1.1 and 2.1.2. Here, we focus on existing human computation approaches.

2.3.1 Text Corpora

Some human computation tasks annotated texts with polarity or emotion labels. Brew et al. [13] invited the users of a news feed to annotate articles as positive, negative, or irrelevant. Each article in the feed came with three links, one corresponding to each polarity value. Workers could annotate news articles by clicking on one of these links. Hsueh et al. [39] asked workers to analyze snippets of blog posts that discussed the election campaigns of some political candidates. Workers indicated if these snippets were positive, negative, both, or neutral with respect to a particular candidate. Melbeek et al. [81] engaged workers to annotate review sentences with their polarity. Chen et al. [18] asked workers to annotate Twitter posts in two phases: first, as to whether or not these were relevant to a particular brand; second, as to whether those marked as relevant contained words expressing opinions about the brand. Snow et al. [106] annotated short sentences with emotions. The authors asked workers to rate sentences according to how much these expressed each of six possible emotion categories. Workers were additionally asked to rate the positive or negative polarity of these sentences. Socher et al. [107] annotated sentences and all their sub-phrases with polarities.

2.3.2 Sentiment Lexicons

Other tasks created sentiment lexicons by annotating the terms in a predefined vocabulary with polarities or emotion categories. Scharl et al. [103] initially designed a game that asked workers to annotate sentences with their polarities. They then adapted this game such that workers could annotate individual words as positive or negative. Hong et al. [36] proposed a game in which workers were grouped in teams of two and asked to agree on the polarities of individual words. Both workers in a team saw the same word, depicted as a box, and had to place it on top of one of three stacks, with positive, negative, or neutral words. If both players agreed on where to place the word, its corresponding block disappeared from the stack. Otherwise, the word remained on the stack for the duration of the game. Mohammad and Turney [83] described a task in which workers had to annotate terms with emotions. For each word, workers had to indicate how representative it was for each of several emotion categories, such as fear, anger, or disgust. Additionally, workers also had to specify how positive and negative the terms were. Makki et al. [73] described an automatic method that generated a context-dependent lexicon containing the polarities of sentiment words in the context of aspects of reviewed items. The authors then asked workers to verify, and correct if needed, the polarities of some of the extracted word combinations. Similar tasks for polarity and emotion annotation were described by Lafourcade et al. [63].

As an alternative to using a predefined vocabulary, some tasks asked workers to annotate all the words in a text passage. For example, Al-Subaihini et al. [4] proposed a game played in rounds. In each round, a worker saw a sentence extracted from an online review. The sentence was split into words, and each word constituted a balloon. Within a limited time, the worker had to assign each balloon to one of four bins corresponding to positive, negative, or neutral polarities, or to entities. According to our understanding, if several consecutive words were placed in the same bin, they were viewed not individually, but as a phrase. When a worker failed to classify a balloon, she lost a life, and losing all lives meant ending the game. At the end of a round, the worker was also asked to assign a polarity label to the whole sentence.

Other tasks created sentiment lexicons by asking workers to select words from the text and label them with polarities. For example, Al-Subaihini et al. [3] proposed a game in which two teams of two workers faced each other in three rounds, the winning team being the first one to pass all of them. In all three rounds, the teams were shown the same review sentence. In the first round, workers were asked to select all the individual words or phrases that had a positive polarity. In subsequent rounds, they were asked to select words that were negative and that represented entities, respectively. In addition, in the last round, workers had to also indicate whether the sentence as a whole was positive, negative, or neutral. The first team to have its members agree on the elements selected was the one that won that round.

Musat et al. [85] also created a lexicon using a game in which workers selected and labeled sentiment words. The game was also team-based, but the roles of the two participants were not symmetric. One of the two workers was shown a product review and was asked to first

2.3. Sentiment Knowledge Acquisition with Human Computation

decide whether the text was positive or negative, then to select an individual word or a short phrase that was the most representative of this polarity. The second worker was then presented with this selection and had to indicate its polarity. If workers agreed on the polarity, they were rewarded with a positive score update. Sintsova et al. [104] obtained a lexicon through a task in which workers read short fragments of text and had to first indicate the emotion category that best characterized the text, then find all the expressions of that emotion in the text. Each expression could be an emoticon, an individual word, or a longer term sequence. As a last step, workers were also asked to enumerate other expressions indicative of the chosen emotion, however, not from the tweet text but based on their personal experience.

Søgaard et al. [108] aimed to extend the feature set of machine learning sentiment classifiers from unigrams to longer word combinations. They remarked that, for a sentence like “*Could have been more favorable*”, its sentiment is correctly captured not by the word *favorable* on its own, but through the phrase *been more favorable*. They also remarked that, in order to cope with cases where these three words are not consecutive in a sentence, a regular expression feature “*been.*more.*favorable*” should be created, thus allowing for more flexible matches in sentences. The authors acquired such features using two sets of workers. In one experiment, experts composed regular expressions with the assistance of software that could indicate how well a particular regular expression correlated with positive and negative texts. In another experiment, the authors recruited workers on a crowdsourcing platform. They showed workers a piece of text and asked them to click on the words or phrases indicative of the text’s sentiment. After the workers’ answers were collected, the authors preprocessed them to extract regular expression patterns.

More Structured Contextual Knowledge Acquisition As we have seen, a raw model of context extends the feature space from individual words to longer word combinations [115, 90]. Therefore, some of these human computation approaches have already acquired contextual knowledge, by allowing workers to select longer sentiment expressions, or by inviting them to click on multiple words in a sentence. However, a more structured and useful model of context should explicitly separate the sentiment expressions from the terms that impact their polarities. We use a paid human computation game that acquires such contextual knowledge with high accuracy. This task is more complex, as it explicitly asks workers to reason about the contexts that can change the polarities of sentiment words.

2.3.3 Fine-grained Annotations

Human computation has also been used to create corpora with more fine-grained annotations, at the level of individual opinion expressions and their targets. Sayeed et al. [102] designed a task in which workers were shown sentences expressing opinions about concepts. Each sentence highlighted a particular concept, as well as several phrases that were likely to convey an opinion about that concept. Workers were asked to discern whether these phrases expressed a positive, negative, or no opinion about the concept that was highlighted. Sauri et al. [101]

described a more complex task. Workers had to analyze full documents and complete several steps, such as locating individual opinion expressions along with their targets, or indicating the polarity or intensity of the opinion.

Improved Fine-grained Annotations There have not been many attempts to acquire fine-grained annotations using human computation. Sayeed et al. asked workers to annotate only opinions and simplified their task by restricting the search space to only a few highlighted words, thus potentially missing some annotations. On the other hand, Sauri et al. allowed workers more freedom, but reported low inter-annotator agreement results. It might be that their task was too complex, in that it asked workers to annotate longer passages of text. It might also be that their aggregated annotations would have had a slightly higher quality when compared to a gold standard, but the authors do not present such results. We use a paid human computation game to acquire fine-grained annotations that have a high accuracy.

2.3.4 Sentiment Knowledge Acquisition with Volunteers

As mentioned in Section 2.2, when a task relies on paid workers, it is completed relatively fast. However, when voluntary workers are involved, there is no clear recipe for luring them to the task and for engaging them to solve it. A solution would be to recruit workers through online advertisements and to incentivize them by enhancing the task with game elements. This has been shown to work for tasks requiring expert knowledge [43]. We inquire whether combining online advertisements with games can also be an effective recipe for recruiting and engaging volunteers to provide commonsense knowledge for sentiment analysis.

3 Knowledge Acquisition for Scalable Sentiment Classification

3.1 Introduction

As a first problem that would benefit from commonsense knowledge, we consider document-level sentiment classification: the problem of automatically inferring whether the sentiments expressed in a text have an overall positive or a negative polarity.

Approaches for document-level sentiment classification fall into two categories. In lexicon-based approaches, the most frequent sentiment words in a language are enumerated along with their polarities, to construct a sentiment lexicon. Documents are classified by matching words from a lexicon, then by predicting the class that is represented by most words. These approaches can be applied broadly, but their accuracy is relatively low. In supervised methods, a machine learning algorithm is applied on an annotated text corpus, for example positive and negative reviews of a particular product category. To keep the learning complexity manageable, the features are generally limited to the most frequent words appearing in the corpus. Based on the feature distribution in the two text classes, these algorithms identify a positive or a negative polarity score for each feature. New documents are classified by summing the weights of the features they contain, thresholding the result at zero. Supervised methods can perform well, but only as long as the domain remains relatively small.

Therefore, a big issue is that sentiment classification methods perform well when reviews are limited to a narrow domain, but decline in accuracy as the domain broadens [126]. The only way to consistently reach a good performance across a broad domain is through a collection of specialized models, each fit for a niche sub-domain. However, as the training domain broadens, the number of specialized models increases and becomes unmanageable. Instead, it would be more convenient to have a single model capable of replicating the aggregate performance of its specialized counterparts. This could even boost performance on sub-domains that target new products and thus do not contain enough training data. However, high-performance broad sentiment models have so far been out of reach.

A main key to this problem lies in how sentiment knowledge is modeled. Both methods rely

on the polarities of individual terms. However, some words are ambiguous and only gain concrete polarities in specific contexts. For instance, the word *small* is positive in the context of a compact digital camera but negative in the context of a vacuum cleaner canister or a hotel bedroom. Other sentiment words are unambiguous, and yet they can have their polarities flipped by neighboring terms. For example, the expression *hardly any good* becomes negative. Since most sentiment lexicons do not refine the polarities of sentiment words in contexts, they consistently perform poorly on broad domains. Conversely, supervised methods can perform well by learning context-dependent polarities on narrow domains, where words appear in only a few contexts. For example, a Support Vector Machine separately trained on cameras, vacuums, and hotels might correctly identify the word *small* first as positive and then as negative. However, on a broad domain, this context approximation shows its limitations. An SVM jointly learning on cameras, vacuums, and hotels will probably not know whether to consider *small* as positive or as negative. This broad model will thus perform worse than its specialized counterparts.

This shows that sentiment classifiers need to incorporate context by considering longer word combinations. In theory, the polarities of longer features could be learned from data. However, this drastically increases the feature space, so a very large corpus would be needed. This is why, in practice, even attempts to restrict to learning word pairs have reported mixed results.

In this chapter, we aim to reliably acquire context using human computation. We acknowledge that, unlike machines, humans can correctly select both sentiment words and their disambiguating contexts, even from very short sentences. For instance, when shown sentences like *This small camera fits in every pocket!* or *Our hotel room was so small we could hardly breathe!*, humans can easily identify that the contexts *camera* and *hotel room* are relevant for the word *small*. Our main contributions can be summarized as follows:

- We show how context can be effectively acquired using human computation.
- We show how human-generated context can be integrated into lexicon and supervised learning methods to obtain classifiers that are applicable on a broad domain.

Context Acquisition

We design our human computation task with several considerations in mind. One concern is acquiring information in a focussed way, while still allowing workers to express complex knowledge. We achieve this by structuring the task in rounds, in which workers read review sentences and submit answers that contain a sentiment expression, a context, and a polarity.

Another concern is recruiting workers and keeping them motivated. Identifying sentiment words and contexts requires cognitive engagement, and this can make workers quickly lose motivation and abandon the task. To increase engagement, we combine enjoyment with payment. As workers submit answers, they are rewarded with: point updates that reflect the

quality of their answers; and with interesting animal puzzles that they gradually solve as they earn points. Once they finish playing, workers also receive payments that are proportional to their final scores. We thus obtain a game played for money, for which we recruit workers on a paid crowdsourcing platform.

A final concern is ensuring quality. Because the context of a sentiment expression can be selected in more than one way, quality assurance by agreement with peers is not possible. Instead, we use a scoring mechanism that steers workers to give answers that have common sense and are novel. We use a scoring model that contains beliefs about the polarities of sentiment words in context. We initialize this model from existing sentiment knowledge and refine it with the workers' answers. We score generously those answers that agree with and strengthen this model, which is what we consider commonsensical and novel, respectively.

Context Integration

We use the game to acquire contextual knowledge that we incorporate into lexicon and supervised methods. We use a dataset organized hierarchically, with multiple narrow domains at the bottom and a broad domain at the top. Lower in this hierarchy, reviews are grouped by electronics (vacuums and cameras), kitchen appliances, and hotels. We acquire a separate context model for each of these domains. At the top of the hierarchy, we target the broad domain with a combined context model.

We study how human-generated context impacts a lexicon method. Lower in our domain hierarchy, we extend this lexicon with the three context models for electronics, appliances, and hotels. We show that each context model substantially improves performance on its corresponding domain. At the top of the hierarchy, we extend the lexicon with the combined context model and show that this further improves performance. We thus show that human-generated context helps the lexicon scale to a broad domain.

We also study how human-generated context impacts a supervised learning method. We analyze specialized and general models trained along the levels of our domain hierarchy. We first show that supervised models using only individual words indeed suffer in performance as they become general. We then show that human-generated context can be integrated to improve over individual words. More importantly, we show that this helps a general supervised model become as powerful as its specialized counterparts. Finally, we show that bigrams can also improve over individual words. However, we show that if we intersect these bigrams with the human-generated features, we find a better subset of context features. This further improves the method and still helps general models be competitive.

The remainder of this chapter is structured as follows. In Section 3.2, we describe the sentiment classification method and how we extend it with context. In Section 3.3, we explain how we design a human computation task for context acquisition. In Section 3.4 we present our experiments, while in Section 3.5 we draw conclusions.

3.2 Document-level Sentiment Classification

We work with online reviews, which come with a main body of text and a gold-standard sentiment label, derived from the reviews' star ratings. To automatically compute a positive or negative label for a review, we use a generic sentiment classification procedure:

1. Define a feature space by processing texts to extract the relevant words.
2. Obtain a sentiment model by attaching polarities to words.
3. Compute a sentiment score by summing word polarities quantified by word frequency in the text.
4. Compute a positive or negative sentiment label by taking the sign of the score, if it is non-zero.

To assess whether this procedure issues a correct classification, we compare the computed sentiment label against the gold-standard label that corresponds to the review.

We focus on both lexicon and supervised learning methods. Lexicons include common sentiment words, to which they attach discrete polarity scores. We consider the sentiment lexicon of Hu and Liu [40]. On our dataset, this model gives a better performance when compared to the General Inquirer and OpinionFinder lexicons. Supervised methods can incorporate any word and identify small continuous polarity scores. We use two supervised models. We consider one feature space with the most frequent words. To avoid overfitting due to over-specialized words, we restrict to the top 1,000 features. We train a linear kernel SVM on a frequency-based bag-of-words representation of the review texts and obtain one supervised model. We also extend these frequent features with the sentiment words in the lexicon of Hu and Liu, and obtain a second supervised model. We train these models using the SMO classifier implementation in Weka¹ [33], by enabling feature normalization and by choosing a relatively small value of 0.1 for the complexity constant C , to help further reduce overfitting. Note that when computing reviews sentiment scores at Step 3, these supervised models also subtract a bias constant.

3.2.1 Bigrams Context

As a raw definition, context is captured by longer features that include two or more words. Therefore, to incorporate context, we can try to extend the feature space to include such word combinations. This would allow to reason about words like *small* not only individually but also as part of expressions such as *small camera*, *small canister*, or *small hotel room*. A classification model could then capture that these features are positive and negative, respectively. While sentiment lexicons do not typically contain polarities for word combinations, we can try to

¹<http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html>

learn these with a supervised method. However, to manage the learning complexity, we only consider bigrams. We obtain another supervised model by extending the feature space of individual words (including the most frequent words and the sentiment words in the Hu and Liu lexicon) with bigrams. We assess bigram utility with the Gain Ratio measure [95] and keep only a certain number of the highest ranked ones. As it is further explained in Section 3.4, we keep as many bigrams as there are word combinations in the context features that we elicit with human computation.

3.2.2 Human-generated Context

We seek to more reliably incorporate context using human computation. We propose a more structured and useful definition of context that is based on two concepts:

- A *phrase* is an expression that can convey sentiment, such as *good* or *small*.
- A *context* is an expression (possibly an empty word), in whose presence the phrase conveys a concrete sentiment, such as *hardly any* or *canister*.

We rely on many human workers to acquire a context model that enumerates phrase and context pairs, along with their positive or negative polarities (Table 4.9 shows sample word combinations selected with human computation).

We use this human-generated model to help sentiment classification become competitive on broad domains. One option is to directly use it to label documents, by applying the generic classification procedure described above. More specifically, at Step 3, we can compute a review sentiment score by summing the polarities of phrase and context pairs quantified by their frequency in the text:

- If the context is not empty, we compute the frequency of a phrase and context pair as the number of times they appear together in the sentences of the review. However, we distinguish the cases in which the phrase and the context are separated by at most three words from other co-occurrence patterns. More specifically, we count the former case as a full occurrence that we quantify with a weight of one. We consider the other cases as partial occurrences that we quantify with a smaller weight, which decreases as the distance between the phrase and the context increases. Similar to Ding et al. [25], this allows us to give less importance to farther away contexts that are less likely to target the phrase in question. Note that, in initial experiments, we found that setting the distance threshold to three gives good results.
- On the other hand, if the context is an empty word, we compute the frequency of the phrase as the number of times it appears in the text outside any of its known contexts.

Therefore, we use the phrases' context-dependent polarities wherever possible and revert to the context-independent ones otherwise.

Chapter 3. Knowledge Acquisition for Scalable Sentiment Classification

A context model does not have full coverage, given that it contains longer word combinations that do not always appear in texts. What we really want is to integrate it into lexicon and supervised methods, which contain individual words on which we can fall back when context features cannot be used. We suggest three ways in which context can be used to extend lexicon and supervised methods. We extend the lexicon by merging it with a context model and using the resulting union to classify reviews. We extend the supervised method with two approaches. One is an ensemble that separately uses the supervised and context models to obtain two sentiment scores, then combines the two to reach a final classification. Another one retrains the supervised method on an extended feature space that includes all the elements in the supervised and context models. Note that this is similar to Kennedy and Inkpen [50], who described how to extend lexicon and supervised methods with negations, intensifiers, and diminishers from a predefined list, and also proposed to combine the two in an ensemble method. We detail our three approaches in what follows.

Sentiment Model Extension

To improve the lexicon method, we apply the extension at the level of the sentiment model, in Step 2. Given a lexicon and a context model, we obtain their union, then use the latter to classify reviews. We merge the two models as follows: for every word in the lexicon model, we add it to the union by pairing it with an empty context component and assigning it the polarity indicated by the lexicon; for every element in the context model, we add it to the union, unless its context is an empty word and the phrase already belongs to the lexicon model. Therefore, the union refines the context-independent polarities in the lexicon model with the context-dependent ones in the context model. We use this union to classify reviews by adding feature polarities quantified by their frequency in the text, as explained above.

Sentiment Score Extension

We cannot apply the previous extension to a supervised model. Lexicon and context models both contain discrete, positive or negative polarities and can thus be merged. However, supervised models contain small continuous polarities that are perturbed when overridden with the discrete ones in a context model. Instead, we improve the supervised method by applying the extension at the level of the sentiment score, in Step 3. We classify a review by separately using the supervised and context models to obtain two sentiment scores. We then combine the two scores, hoping to rectify some of the errors produced by the supervised model. To make the two scores compatible, we use a parameter that scales down the discrete one obtained with the context model. A good starting point to choose a value for this parameter is the average polarity magnitude in the supervised model. In our initial experiments, we find that a value of 0.08 works well.

Feature Space Extension

The previous approach integrates context by combining two sentiment scores separately obtained with a supervised model and with a context model. We expect that context can be more effective if it is integrated in the training process, when we can rely on the SVM to find suitable polarity scores for all the features involved. Therefore, to further improve the supervised method, we propose to apply the extension at the level of the feature space, in Step 1. We extend the feature space so that it includes elements from both the supervised and the context models. We then retrain an SVM on this extended feature space and obtain a new supervised model.

To represent a review in the extended feature space, we take two approaches. When we want to extend a supervised model containing only individual words, we obtain a feature space that simply unites all the words and word combinations in the supervised and context models. We represent each review sentence as follows:

- We first find all the elements in the context model for which both the phrase and the context appear in the sentence. For every match, we mark the words involved and output a feature that concatenates the phrase and the context.
- We then find all the words in the supervised model that appear in the sentence and are still unmarked. For every match, we output a feature capturing that word.

For instance, let us assume we have a sentence: *I hate this vacuum, it has a very small canister.* A supervised model that contains the words *hate*, *very*, *small*, and *canister* will interpret the sentence as *[hate][very][small][canister]*. However, when we extend this feature space with a context model that contains the phrase *very small* in the context *canister*, the resulting supervised model will interpret this sentence as *[very small canister] [hate]*.

When we want to extend a supervised model that also contains bigrams, the approach above can output many superfluous features that often overlap with one another. Instead, we obtain a feature space that contains the individual words and the bigrams in the supervised model, but we restrict only to those bigrams that also appear in the context model. We represent each review sentence as follows:

- We first find all the bigrams in the supervised model whose both constituent words appear in a phrase and context combination from the context model. We output every such bigram that appears in the sentence.
- We then find all the words in the supervised model that appear in the sentence. For every match, we output a feature capturing that word.

For example, a supervised model that contains the features *hate*, *this vacuum*, *a very*, and *small canister* will interpret the sentence from before as *[this vacuum][a very][small canister]*

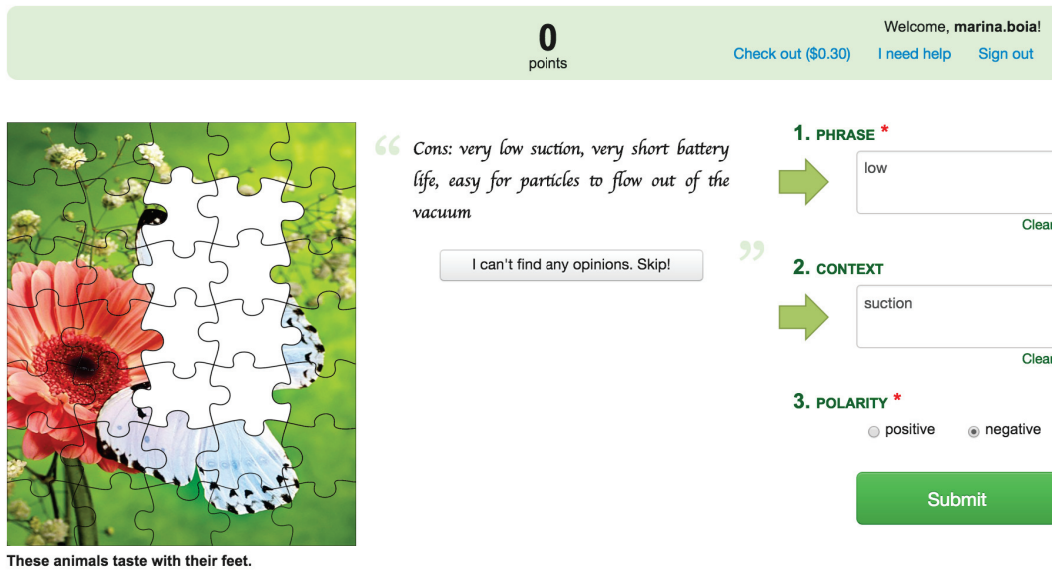


Figure 3.1: Context acquisition. Main game interface

[hate]. However, we can intersect the bigram space with a context model that contains the phrase *very small* in the context *canister*. The resulting supervised model will interpret this sentence as *[small canister] [hate]*, thus effectively pruning some of the irrelevant bigrams.

3.3 Human Computation Task

To acquire a context model, we ask workers to submit answers that contain the polarities of phrase and context combinations. We design our human computation task with several considerations in mind:

- We need to obtain information in a structured way, while still allowing workers to express complex knowledge.
- We need to recruit workers and motivate them to invest effort and stay with the task.
- We need to ensure that workers understand the task and are qualified to do it.
- We need to control the quality of answers.

3.3.1 Task Structure

To obtain information in a focussed way, we structure our task in rounds. In each round a worker needs to formulate a judgement regarding the polarity of a phrase and context pair. To still allow workers to express complex knowledge, we do not restrict them to labeling a fixed, predefined vocabulary of word combinations. Instead, we give them full flexibility to create

complex answers. Therefore, each round displays a sentence extracted from a review. From this sentence, a worker needs to construct an answer in three steps:

1. *Phrase*: selecting a phrase for the sentence.
2. *Context*: optionally selecting a context for the phrase.
3. *Polarity*: labeling the phrase and context pair with a polarity.

When a worker submits her answer, she starts a new round.

3.3.2 Worker Motivation

We encourage workers to focus on submitting answers that would be of most use to a sentiment classification model: answers that contain phrases along with contexts that have an impact on their polarities. This requires them to reason more elaborately, as not all contexts can change the polarity of a phrase. Therefore, the task requires cognitive engagement, which is why workers are likely to quickly lose their motivation and abandon in. However, it is not clear if using only extrinsic motivators such as financial rewards can sufficiently engage users. In some previous experiments, we obtained low quality results with a review polarity annotation task in which we engaged colleagues with prizes.

To increase engagement, we target both intrinsic and extrinsic motivation, by relying on enjoyment and payment. We entertain workers with point rewards and puzzles. After each answer submitted, a worker receives a point reward that reflects the usefulness of her answer. Moreover, by submitting answers, the worker solves puzzles that point to animals with intriguing properties or behavior. Each answer unlocks hints for these puzzles, which consist of gradually revealing a picture that conveys the puzzle's solution. When a worker collects all the hints for a puzzle, she gets further explanations about the portrayed animal and sometimes links to Web pages that give even more details. Once they finish the task, workers receive financial rewards that are directly coupled with their scores. By combining enjoyment with payment, we obtain a game played for money, like poker or other card games. The interface of the game is displayed in Figure 3.1². For the game art, we used:

- Butterfly image by Douglass Sprott³ (altered), under Creative Commons licence⁴.
- Jigsaw puzzle image by Arpop⁵.
- Animal fact: butterflies taste with their feet⁶.

²The games in this thesis were implemented in the Java Play framework: <https://www.playframework.com>

³<https://www.flickr.com/photos/dugspr/5732623724>

⁴<https://creativecommons.org/licenses/by-nc/2.0>

⁵<http://fssp-arpop.blogspot.ch/2009/11/jigsaw-puzzle-templates.html>

⁶<http://biointerestingfacts.blogspot.ch/2007/01/butterflies-taste-with-their-feet.html>

1 HOW TO FIND PHRASES

Phrases are words and word combinations that express positive or negative opinions about something (e.g. *excellent*, *dreadful*).

Your task: In the sentence below, **find a phrase** that is negative

1. First, **highlight the phrase in the text**. Note that more than one word is allowed.
2. Then, **click on the first green arrow** to save it in the phrase answer box.
3. Finally, click on the 'submit' button to check your answer.

“ *This product is disappointing.* ”

1. PHRASE

→ disappointing

Clear

2. CONTEXT

→

Clear

3. POLARITY

positive negative

Figure 3.2: Context acquisition. Tutorial quiz that asks workers to identify a phrase with a negative polarity

3.3.3 Tutorial

To ensure that workers understand the task and are qualified to do it, we create a tutorial that explains the basic principles. The tutorial uses both text instructions and interactive quizzes. It starts by explaining the concepts of phrase, context, and polarity. Each explanation is succeeded by a quiz asking workers to solve part of a normal game round. The first quizzes ask workers to identify a phrase with a given polarity (Figure 3.2). The subsequent quizzes invite workers to identify contexts that make a given phrase first positive and then negative (Figure 3.3). Next, the tutorial challenges workers to solve a couple of more quizzes emulating the full rounds in the game. Finally, the tutorial explains the round-based structure of the game, the scoring mechanism, and the animal puzzles (Figure 3.4). Workers cannot graduate the tutorial unless they correctly solve all the quizzes. We thus ensure that only those workers with a good understanding move on to play the game.

3.3.4 Quality Assurance

The tutorial implicitly influences quality before the game, by allowing only the workers that have gained a good understanding to proceed to the real task. The scoring mechanism controls quality during the game, by encouraging workers to submit useful answers. As an additional safety measure, we allow workers to submit at most 200 answers, to prevent them from submitting doing sloppy work as a result of fatigue. Moreover, if after fifty answers, their average score falls below a predefined threshold, we again stop workers from solving further

2 HOW TO FIND CONTEXTS

Contexts are words and word combinations that can influence the polarity of phrases. For instance, in the contexts *beer* and *pizza*, *cold* is first positive, then negative.

Your task: In the sentence below, **find a context** that makes *small* positive:

1. First, **highlight the context in the text**. Note that more than one word is allowed.
2. Then, **click on the second green arrow** to save it in the context answer box.
3. Finally, click on the 'submit' button to check your answer.

“

This blender came at an *incredibly small price*.

”

1. PHRASE

➔

Clear

2. CONTEXT

➔

Clear

3. POLARITY

positive negative

Figure 3.3: Context acquisition. Tutorial quiz that asks workers to identify a context which makes a phrase positive

rounds (we come back to choosing this threshold when we describe the scoring mechanism below). In both cases, workers are seamlessly notified that no more rounds are available to them. We also control quality after the game, by first filtering out the workers that did a bad job, then by removing the remaining bad answers.

Scoring Mechanism

We use a scoring mechanism that encourages workers to submit useful answers. We judge the usefulness of an answer using two criteria: whether it has common sense and whether it brings new knowledge. We assess that an answer is commonsensical if it is consistent with a scoring model that aggregates the workers' activity in the game up to that point: this means that the answer agrees with the common judgement of previous workers. We establish that an answer is novel if it has a great impact on the scoring model: this means that the answer is submitted early on and that it contains a phrase that requires a context for polarity clarification, along with such a disambiguating context. We compute score rewards by adding an agreement score with a novelty score. Even if our scoring model contains some initial mistakes, workers do not know where these occur. They thus need to consistently provide useful answers to maximize their score. As a result, any initial errors in the model should be corrected over time.

Our scoring model contains beliefs about the polarities of phrase and context combinations. This model attaches a Beta distribution [32] to each phrase and context pair. From this

4 HOW TO PLAY THE GAME

In this game, you play with opinions and their contexts. As you progress, you unlock cool facts about animals.

4.1 The rounds

You play in simple rounds. In each round, you see a sentence from an opinionated text. From this sentence, **you need to construct an answer**. To do so, you have to:

1. Highlight a phrase, then click on the first green arrow.
2. A context may be necessary to clarify the polarity of the opinion phrase. If so, highlight a context in the text, then click on the second green arrow.
3. Select whether the opinion in context is positive or negative.
4. Submit your answer.

If you're unsure about how to construct the answer, you can skip to the next round.

4.2 The points

You receive points on each answer. For scoring, we compare you with previous players. **You earn the most points when:**

1. You agree with most players on the polarity of an opinion phrase in context.
2. You identify opinion phrases that need contexts for their polarity to become clear.
3. You are among the first to identify a certain combination of an opinion phrase and context.

4.3 The puzzles

At any given point, a puzzle is assigned to you, pointing out an awesome fact about an animal. As you submit answers, you unlock parts of this puzzle. When you completely unlock a puzzle, you get a new one.

4.4 The end of the game

You can stop the game at any point, by clicking on the 'check out' button. Otherwise, the game will end when you've exhausted all the rounds.

Figure 3.4: Context acquisition. Tutorial explanation of the game rounds, scoring, and puzzles

distribution, we can estimate the probabilities that, in that context, the phrase has positive and negative polarities, respectively. We initialize these Beta distributions using corpus statistics and several sentiment lexicons. To account for pairs containing phrases and non-empty contexts, we use the word pairs that co-occur in the sentences of our review corpus, and we attach a corresponding Beta distribution to each. We initialize a distribution based on the difference between a word pair's frequencies in positive and negative documents, respectively. When the word in the pair playing the role of the phrase also appears in a sentiment lexicon, we complement the corpus frequencies with its polarity score. To account for pairs containing phrases and empty contexts, we use the individual words in our corpus, and we attach a Beta distribution to each. We initialize these distributions in a similar manner. We use a Bayesian update process to incorporate incoming answers into these Beta distributions. As a result, the positive and negative probabilities assimilate the fractions of positive and negative answers, respectively.

For a new answer, we compute an agreement score that assesses if it is commonsensical. We set this agreement score highest if the answer agrees with the scoring model early in the game, since this improves the model's confidence. We set the agreement score lowest if the answer contradicts the scoring model early in the game, since this damages the model's confidence. Finally, we assign a low-to-medium value to the agreement score when the answer comes later in the game, because, at that point, it has a smaller impact on the scoring model's confidence. To assess the model's uncertainty in the polarity of a phrase and context, we compute the entropy over the pair's corresponding polarity distribution. The answer decreases this entropy if it agrees with the model and vice versa. Moreover, the answer produces bigger changes

in entropy earlier in the game. We thus obtain the agreement score by (piecewise) linearly mapping the updates in entropy to the agreement score interval.

To further reward the answer, we also compute a novelty bonus. This reflects whether the answer contains a phrase whose polarity can fluctuate, along with a non-empty context. Specifically, we focus on the phrases that are ambiguous by themselves, such as *small*. As an indicator for the phrases's ambiguity, we use the scoring model's uncertainty in its out of context polarity. If the context is an empty word, we set the novelty score to zero. Otherwise, we assess the phrase's ambiguity by computing the entropy over the polarity distribution attached to it. We obtain the novelty score by (piecewise) linearly mapping this entropy to the novelty score interval. Note that, as it is designed, this novelty score does not generously reward the answers containing contexts that flip the polarities of unambiguous phrases, given that these should have a low entropy according to the scoring model. However, if a context genuinely flips the polarity of an unambiguous phrase, an answer should still be generously rewarded through the agreement score.

We reward the answer with a total score update summing the agreement score and the context novelty bonus. Because we do not want to encourage answers that are submitted only once, we give a bigger importance to agreement by setting the maximum agreement score higher than the maximum novelty score. In initial experiments, we choose values of forty and ten points, respectively. More specifically, in terms of the agreement score, we map the answers that increase entropy to the interval $[0, 10]$ and those that decrease entropy to the interval $[10, 40]$. In addition, we notice that, based on the corpus statistics that initialize the scoring model, even the unambiguous sentiment words tend to have a relatively high entropy. Therefore, in terms of the novelty score, we heuristically set an entropy threshold to 0.9. We reward this bonus to answers attaching contexts to words that meet this ambiguity threshold. In these cases, we simply multiply the entropy with the maximum novelty score of 10.

Given our choice of scoring parameters, Figures 3.5 and 3.6 show scenarios of how the two score components vary in time, as workers submit answers (both of these scenarios are sampled from one of our game launches, further described in Section 3.4). The first example is relatively simple, and shows how the score updates evolve as workers keep indicating that the word *good* is positive outside any context. Given that these answers do not contain a context, they do not receive the novelty bonus. In addition, we can see that, at the beginning, the agreement scores are very high (the first answer is rewarded with the maximum agreement score of forty), but as workers keep indicating the same polarity, the scoring model becomes more confident, so the agreement scores steadily decrease. The second example is more complex, in that it shows how the scoring updates evolve as workers indicate polarities for the phrase and context combination *small room*. Most of the time, workers agree that this expression is negative, which is why we see the same behavior in the evolution of the agreement scores. There is one exception, when a worker specifies a positive polarity, which is penalized with a low agreement score. In addition, for the most part, the scoring model is ambiguous about the polarity of the phrase *small* outside any context, which is why these answers are

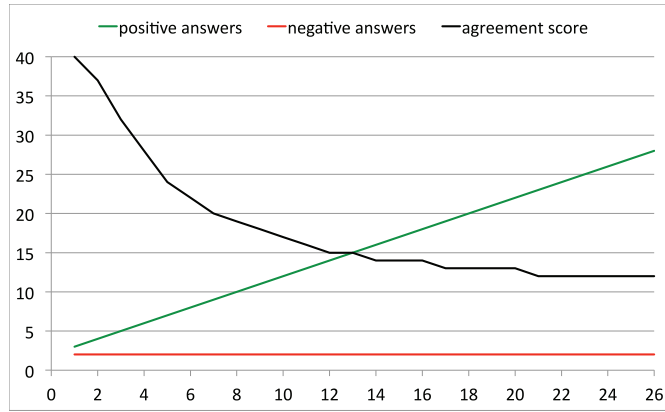


Figure 3.5: Context acquisition. A simple example of how the score updates vary in time as workers keep submitting the answer (*good, , positive*)

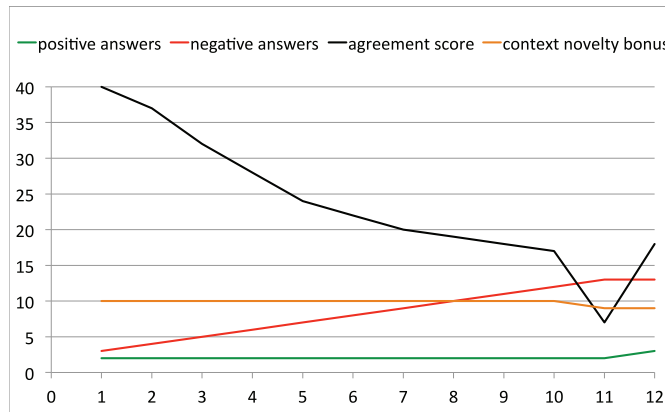


Figure 3.6: Context acquisition. A more complex example of how the score updates vary in time as workers mostly submit the answer (*small, room, negative*) with some exceptions stating the reverse polarity

also rewarded with the maximum novelty bonus. At some point, one worker indicates that this word is positive by itself, which slightly decreases the model’s confusion about the term. This is why, toward the end, we see the novelty bonus decreasing by one unit.

Finally, as it was previously indicated, an additional quality assurance measure that we take during the game is to stop workers if, after at least fifty answers submitted, their average score happens to fall under a predefined threshold. Given our choice of parameters for the scoring mechanism, we heuristically set this threshold to twenty. This is because we require that, on average, workers should submit answers that decrease the scoring model’s uncertainty (entropy), and hence earn an agreement score of at least ten. We also want answers that attach a context to an otherwise ambiguous phrase, which by itself has a high entropy according to the scoring model. This means that such answer should earn a context novelty bonus or nine or ten points. Of course, the score threshold does not exclude answers that do not include a context component. However, because they do not get the novelty bonus, such answers

should on average earn an agreement score of at least twenty. This means that answers without contexts are acceptable as long as they are submitted relatively early in the game (they are not repetitions).

Worker Filtering

We filter out the answers of lazy workers by measuring their performance on gold-standard game rounds. We define several such rounds that we interleave with the regular rounds in the game. The gold rounds show simple sentences with only a few acceptable answers. For each worker, we establish the number of answers she submitted in gold rounds and how many of those answers belong to our set of acceptable, gold-standard answers. We consider the worker's performance is acceptable if she gave correct answers in the majority of the gold rounds that she attempted to solve. We tend to set this accuracy threshold to 80%.

Answer Filtering

After we eliminate the bad workers, we aggregate the remaining answers to obtain a human-generated context model. We consider each phrase and context that one or more workers combined in their answers, and we add it to the context model. To each pair, we attach the majority polarity that results from the workers' activity. From the resulting context model, we remove the phrase and context combinations that lead to many classification errors. Using a corpus of training reviews, we first classify documents with the Hu and Liu lexicon alone. We then combine this lexicon with our context model using the sentiment model extension, and we reclassify the training reviews. For each review, three scenarios can happen: the context model fixes an error of the sentiment lexicon; the context model harms a correct classification of the lexicon; or it neither helps nor harms the output of the lexicon. For each phrase and context pair in our context model, we keep track of improvement and error counts, which we increment when the first or second scenarios occur, respectively. We use these counts to remove the bad elements that damage the performance of the Hu and Liu lexicon.

We delete the bad elements in four iterations. In each iteration, we classify documents and generate the improvement and error counts. We then choose a heuristic for pruning elements. In the first two iterations, we delete the elements with error counts above a predefined threshold. These elements tend to have incorrect polarities and are also very frequent in the corpus. Because they produce errors in many documents, they also add noise to the error counts of the elements they co-occur with. This is why we aim to remove them first. In the following two iterations, we remove the remaining elements that have high error counts as well as the elements whose error counts exceed the improvement counts. In our experiments, we choose the error count threshold in correlation with the size of the training corpus. Generally, this parameter ranges between 100 and 300.

3.3.5 Worker Recruitment

We recruit paid workers on Amazon Mechanical Turk. We create a small HIT that invites workers to play our game. This HIT briefly presents the purpose of our game, then explains: how workers can access the game site; and that, to receive their payment, workers should come back to the HIT page and submit a validation code received at the end of the game. Our HIT also inquires how workers perceived the game. At the end of the game, besides submitting their validation code, workers can optionally fill in a survey with multiple choice questions, asking about: the quality of the game (very poor to very good), its complexity (very difficult to very easy), whether they enjoyed it (no or yes), or how often they would play the game (never to always). The survey also invites workers to write comments or suggestions for improvement.

We reward workers with payments proportional to their final scores in the game. This is possible, given that the platform allows for workers to be rewarded with base payments as well as with bonuses: the base payment is the fixed reward that a recruiter advertises for the successful completion of a HIT; a bonus payment is not fixed and can be optionally received when a recruiter decides to further reward a worker for the quality of her answers. Therefore, the base payment can be complemented with bonuses to pay workers with the full amounts earned in the game.

3.4 Empirical Results

We tested the human computation design using review data and paid workers. We launched the game three times, using sentences from the electronics, appliances, and hotels domains, respectively. We recruited roughly 1,700 workers, who provided a total of 143k answers. After quality control, we compiled these answers into a context model containing 40k phrase and context combinations.

We present results about the performance of the techniques we developed, structured around several major conclusions that they support.

3.4.1 Dataset

We illustrated the need for context in a scenario with multiple narrow domains that could be hierarchically organized from the very specific to the most general. This scenario allowed us to evaluate lexicon models on the various narrow domains, and we hypothesized that these would show a consistently poor performance, due to the lack of context. This also allowed us to train supervised models with varying specialization degrees, along the levels of the hierarchy. We hypothesized that supervised models that did not incorporate context would perform well at the lower levels of this hierarchy, but that they would degrade towards the top level, as the training domain became broader. Finally, we hypothesized that, by incorporating context features, lexicon and supervised models would become competitive on broad domains.

category	training size	testing size
amazon dataset		
vacuum cleaner categories		
hand held	2,204	1,052
canister	2,222	1,076
stand up	7,828	3,672
robotic	1,898	898
digital camera categories		
point and shoot	28,820	7,106
compact	276	62
dslr	1,512	292
video	11,106	3,062
kitchen appliance categories		
blender	5,920	3,074
coffee machine	16,590	8,488
oven	6,084	3,032
grill	2,378	1,192
tripadvisor dataset		
hotel	10,002	3,998

Table 3.1: Context acquisition. Sizes of the training and test sets for each category

To the best of our knowledge, existing corpora for sentiment classification do not easily allow this desired hierarchical structure. We thus created a new dataset with Amazon and Tripadvisor reviews. From Amazon, we used reviews that described twelve categories of vacuums (hand held, canister, stand up, robotic), cameras (point and shoot, compact, dslr, video), and kitchen appliances (blender, coffee machine, oven, grill). From Tripadvisor, we used hotel reviews obtained from Musat et al. [86].

We noticed that some of the Amazon reviews appeared more than once. This is because some products are very similar (e.g. similar camera models differing only in color) and Amazon displays the same review texts for all of them. Moreover, some customers post the same review multiple times, on the pages of related goods produced by the same company. On the other hand, we saw that this was not an issue with the Tripadvisor hotel reviews. Therefore, we had to fix this problem by removing the duplicate Amazon texts.

We used the reviews' star ratings to distinguish between the positive and the negative texts. Both Amazon and Tripadvisor reviews have ratings in the range of one to five. We considered that reviews with ratings above three were positive, that those with ratings below three were negative, and we dropped the reviews with a rating of three. For each category, there were substantially more positive reviews than there were negative. We thus randomly dropped some of the positive reviews, such that the positive and negative classes were balanced.

Finally, we split the datasets corresponding to each category into a training set and a test set.

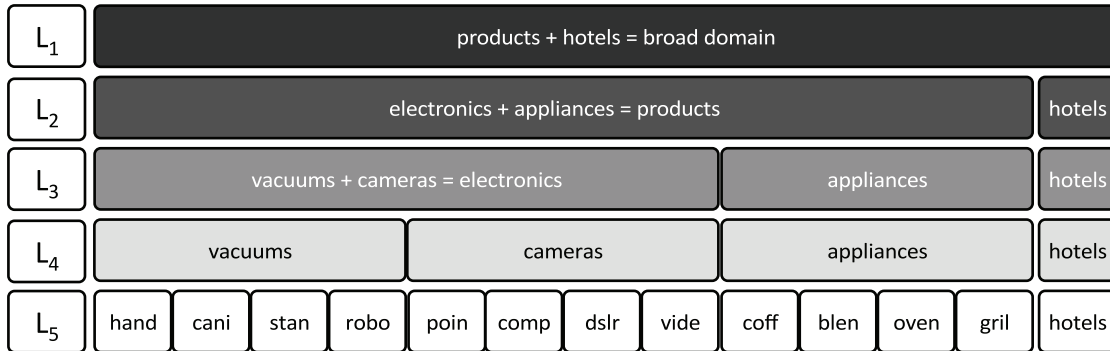


Figure 3.7: Context acquisition. Hierarchical structure of the training set

domain	workers	answers	cost
electronics	970	78,000	\$1,500
appliances	380	31,300	\$573
hotels	380	33,800	\$565

Table 3.2: Context acquisition. Details of the game launches

From the test sets, we removed two camera categories (compact and dslr cameras), which had become too small after duplicates removal. In total, the training and test data consisted of roughly 97k and 37k reviews, respectively (Table 3.1).

We organized the training data in a hierarchy with five levels, in a bottom-up fashion (Figure 3.7). At the lowest level resided thirteen narrow domain datasets, corresponding to the twelve individual product categories and to hotels. At the middle level, we had three datasets corresponding to electronics (vacuums and cameras combined), appliances, and hotels. At the top level, there was one dataset that included all domains.

3.4.2 Task Setup

We acquired context features by launching our game on Amazon Mechanical Turk. We first launched the game with sentences from electronics reviews, then moved on to hotels and appliances. To construct the rounds for the first two game launches, we used a small set of reviews that was separate from the training and test sets. For the third game launch, we used some reviews from the training corpus. In general, we avoided using texts from the test set. We tried to populate the game with sentences selected from reviews that were more difficult to classify. In addition, we made sure that the selected sentences were likely to express sentiments, such that workers did not have to skip a lot of rounds. After a few initial trials, we converged to a payment scheme rewarding each worker with: a base payment of \$0.2 - \$0.3 if she successfully graduated the tutorial and accumulated 100 points in the game; and with a bonus of \$0.05 - \$0.06 for every 100 points that she earned in the game after that.

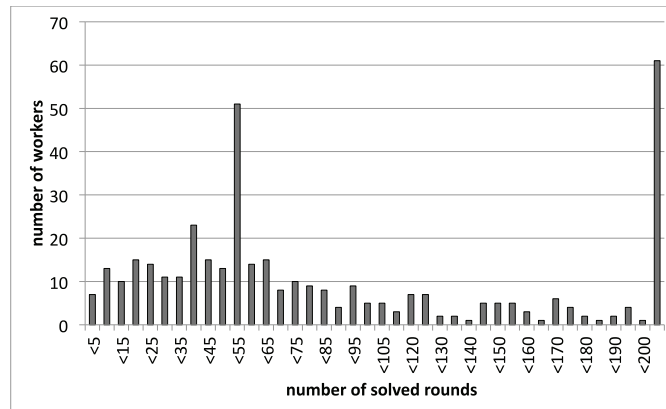


Figure 3.8: Context acquisition. Distribution of workers in terms of the number of solved rounds (for the hotels game launch)

Context can be easily acquired with human computation, at a reasonable price per domain

For the appliance and hotel domains, it was sufficient to recruit about 400 workers, who contributed with approximately 30k answers, and whom we paid with approximately \$600. For the electronics domain, which was larger as it included both vacuums and cameras, the statistics are roughly double (Table 3.2). From these three launches, we learned that, for a domain of the size of hotels, it suffices to recruit 400 - 500 workers. These will submit eighty to ninety answers on average, stopping when they have earned a payment of about \$1.5. (We also looked in more detail at the distribution of workers in terms of the number of solved rounds - submitted answers. In Figure 3.8, we show this distribution for the hotels game launch - the distributions corresponding to the electronics and appliances game launches had similar shapes. We remind that, once workers meet the fifty answers mark, we lock them if their average score happens to fall below a predefined threshold. We also prevent workers from contributing with more than 200 answers. This explains the two bumps that appear in the distribution of workers. The fact that we lock workers, as well as fatigue that sets in as workers stay in the game for a longer time, might also explain the decreasing trend we notice in the workers' distribution, starting from fifty answers onwards.) The outcome will be a set of 30k - 40k answers, for a total cost of \$600 - \$750. These statistics show that, with our task setup, context features can be easily acquired at a reasonable price per domain.

Games are welcome on paid crowdsourcing platforms

The game quality survey showed that 17% of the workers found it to be average, whereas the majority thought the game was good (Figure 3.9). Moreover, 35% thought the complexity was average and half of the workers found that the game was easy. Consequently, most of the workers enjoyed the game and said they would frequently play it. Some workers also wrote explicit comments saying that they found the game interesting and even suggested new animal puzzles. The outcome of this survey was also backed up by the workers' activity in the

game, who on average played well beyond the base payment conditions.

Regarding the interplay between extrinsic and intrinsic motivation, prior research [100] has shown that the former can hinder the latter, at least for tasks that participants would have solved out of interest alone. However, we argue that, even though we combine these two types of motivation, there is no interference between them. This is mainly because we recruit workers on a crowdsourcing platform, which implies that: extrinsic motivation is tied to the platform's culture and that workers expect financial rewards by default. While this means that workers who solve crowdsourcing tasks are strongly driven by extrinsic motivation, precisely because payment is a default, it does not impact the enjoyment. On the contrary, we believe that, when invited to solve two tasks in the same reward range, one fun and one not, most workers would naturally choose the former. Because most tasks on Amazon Mechanical Turk do not include an enjoyment component, our game has an extra advantage over other tasks in the same price range. Moreover, because our task is more fun than the norm, it might also convince workers to play the game beyond the payment they originally had in mind. We thus believe that, for our particular task setup, extrinsic and intrinsic motivation work together. This intuition is backed up by our survey results, which show that the workers received the game positively. This shows that gamified tasks are a welcome addition on Amazon Mechanical Turk, where a lot of tasks do not include an enjoyment component.

3.4.3 Context Statistics

Through these game launches, we acquired three sets of context features, one for each domain at the middle level in our hierarchy. After each launch, we applied quality assurance to remove the bad workers and the remaining bad answers. We aggregated the remaining answers to obtain three context models with 16.6k, 12.7k, and 12.5k elements, respectively. At the middle level of the hierarchy, we separately used these three context models. At the top level of the hierarchy, we combined them in an overall context model with 40.3k elements. On average, the phrase component contains two words, whereas the context component has a length of 1.4 words. Most elements consist of longer word combinations and 760 items are individual words (these elements consist of a one-word phrase and an empty context).

We remarked that, despite our efforts to structure context, the notion is somewhat subjective. For example, given a sentence like *The camera has a short battery life*, a lot of workers will indicate that the phrase *short* is negative in the context *battery life*. However, there will also be some workers who will indicate that the phrase *short battery life* is negative, leaving the context component empty. In total, there are roughly 2,300 elements in the overall context model for which the phrase component contains more than one word, whereas the context component has been left empty. Given the above mentioned variation in how workers understood and selected the context component, we considered that these longer word combinations also incorporated context to some extent (according to the raw definition of context).

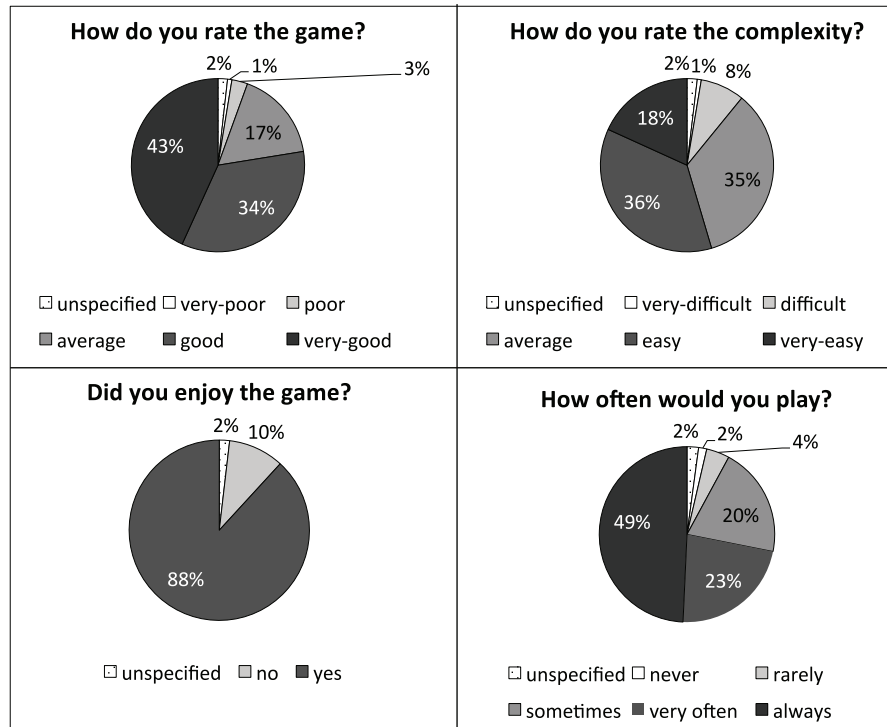


Figure 3.9: Context acquisition. Summary of the game quality survey

3.4.4 Context and the Lexicon-based Method

Context helps the lexicon method become competitive on a broad domain

We studied how human-generated context impacts the lexicon method when integrated through the sentiment model extension. We started by evaluating the Hu and Liu lexicon alone. At the middle level of our domain hierarchy, we extended this lexicon with the three context models for electronics, appliances, and hotels, then we evaluated it on the corresponding domains. At the top level of the hierarchy, we extended the lexicon with the combined context and evaluated on all domains. Because the human-generated context also contains a few individual words, we wanted to assess how much of the improvement was due to these features, and how much was due to the longer word combinations. We also performed intermediate experiments, in which we extended the sentiment lexicon with only these individual words. We evaluated on each category and recorded the average error over the vacuum, camera, and appliance categories, respectively. We then recorded the overall average error over vacuums, cameras, appliances, and hotels.

The Hu and Liu lexicon alone gave an average error of 31.84% (Figure 3.10). At the middle level of the domain hierarchy, the three separate context models decreased the lexicon's error to 15.13%. At the top level, the combined context model further decreased the error to 12.99%. According to a two-tailed paired t-test, the improvements achieved at both levels were statistically significant on all categories (Table 3.4). When testing the intermediate effect of

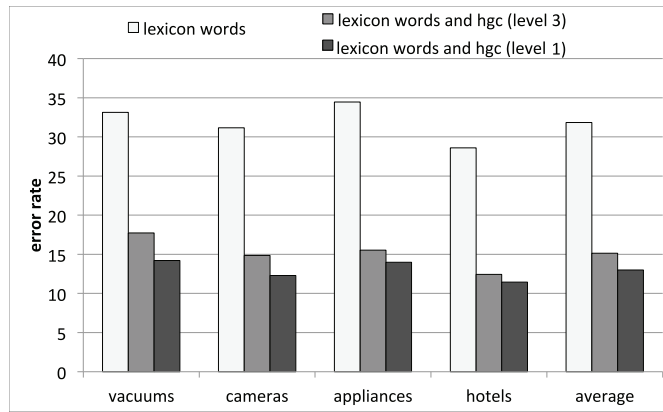


Figure 3.10: Context acquisition. Error of the lexicon with the sentiment model extension. The Hu and Liu lexicon alone, then extended with the separate human-generated context (hgc) at level 3, and finally with the combined hgc at level 1

the individual words in the human-generated context, we observed that, at the middle and top levels, the lexicon's error was reduced to only 27.96% and 28.22% respectively. The further improvements achieved through adding the longer word combinations were statistically significant at both levels of detail, on all categories (Table 3.5).

Therefore, the Hu and Liu lexicon had a relatively low performance. Next, it was only marginally improved when it was extended with the individual words in the human-generated context. Finally, it was substantially improved when it was extended with all the context features. Moreover, the combined context at the top level not only replicated, but further improved the separate effects of the context models at the middle level. This shows that human-generated context greatly improved the lexicon method and helped it scale to a broad domain.

3.4.5 Context and the Supervised Learning Method

Supervised models that do not include context degrade as they become more general

To showcase the need for context in the supervised method, we studied individual words models trained at different levels of our domain hierarchy. We started at the bottom level, where there were thirteen narrow domains. For each domain, we trained two models: one using the most frequent words and another additionally using words from the Hu and Liu lexicon. We continued at level 4, where there were four domains (vacuums, cameras, hotels, and appliances). For each domain and feature set, we again trained a supervised model. We repeated this up to the top level, where there was one broad domain. For each hierarchy level and feature set, we evaluated the resulting supervised models on the test sets falling within their scope. For instance, at level 4, we evaluated a vacuums model on the four test sets corresponding to the individual vacuum categories.

We observed that the model based on frequent words decreased its error when moving from

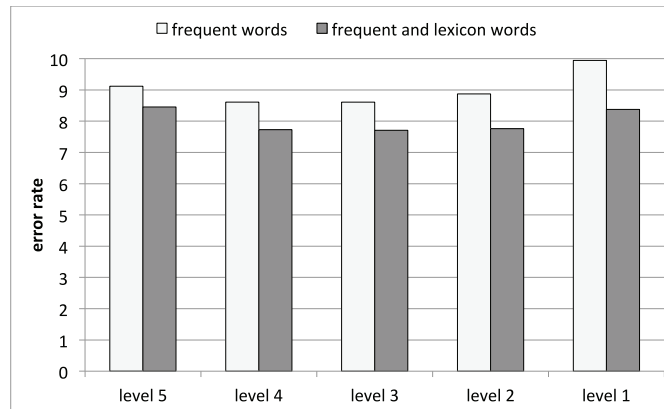


Figure 3.11: Context acquisition. Error of supervised models trained at the five hierarchy levels, from level 5 (individual categories) to level 1 (products and hotels combined)

level 5 to level 3, at which point it started to harm performance, up to level 1. The model using both the frequent words and the lexicon words showed a similar but less pronounced behavior (Figure 3.11). Therefore, it seems that these models benefited when moving from level 5 (where individual categories resided separately) to level 3 (where electronics, appliances, and hotels resided separately) from more training data coming from merging related product categories. However, when we started merging datasets with more obvious differences, these models gradually degraded their performance. This confirms that a supervised model that does not model context cannot be competitive on broad domains.

Context helps broad supervised models become as powerful as specialized ones

We studied how human-generated context impacts the supervised method when integrated using the sentiment score extension. We extended the supervised model based on frequent and lexicon words. At the middle level in our hierarchy, this model produced a minimum error of 7.71%. However, at the top level, this model increased its error to 8.38%. We used the sentiment score extension to complement this latter model with the combined context for all domains. This decreased the error to 7.75%. The improvement was statistically significant on one camera and one appliance categories and on hotels (Table 3.6). We also performed an intermediate experiment where we complemented the supervised model with only the individual words in the combined context. This actually harmed the performance of the supervised model, increasing the error to 8.67%. Further adding the longer word combinations brought statistically significant improvements on two camera and three appliance categories and on hotels (Table 3.7). This shows that the improvement recorded when complementing with the full context model was due to these longer features. Therefore, even if it was not integrated in the training process, human-generated context improved the general supervised model and made it perform as well as the latter's specialized counterpart.

We also studied how context impacts the supervised method when integrated using the feature

space extension. At the middle level, we extended the three individual words supervised models with the three separate context models for electronics, appliances, and hotels, respectively. This decreased the error from 7.71% to 7.35% (Figure 3.12). The improvement was significant on two appliance categories. We also tested the intermediate effect of using only the individual words in these context model, when we recorded an error rate of 7.68%. Further adding the longer word combinations brought statistically significant improvements on the same appliance categories.

At the top level, we repeated this procedure and extended the individual words supervised model with the combined context. This decreased the error from 8.38% to 7.31%. The improvement was significant on: two vacuum and three appliance categories; hotels (Table 3.8). When we extended the supervised model only with the individual words in the combined context, we recorded an error rate of 8.19%. Further adding the longer features brought improvements that were statistically significant on the same categories (Table 3.9).

Therefore, context improved the individual words supervised models at both levels of detail, and the error decrease was mostly due to the longer word combinations. At the top level, this improvement was greater than the one we obtained with the sentiment score extension. More importantly, unlike the individual words supervised models, which decreased in performance when they became broader, the supervised models that incorporated context performed comparably in both their specialized and general versions. This means that human-generated context helped the supervised method scale to a broad domain.

Bigrams also improve the supervised method. However, intersecting them with the human-generated context makes them more efficient and still helps the method scale

We also studied how human-generated context compares to bigrams. At the middle level, we extended the three individual words supervised models with the three human-generated context models for electronics, appliances, and hotels, respectively. Then, for each of the three domains, we replaced the human-generated features with bigrams. We used as many bigrams as there were longer word combinations in the corresponding human-generated context. Finally, for each domain, we intersected the bigrams with the corresponding human-generated context model. At the top level, we repeated the same steps.

At the middle level, the human-generated context and bigrams gave errors of 7.35% and 7.06%, respectively. Intersecting the two types of context decreased the error to 6.27% (Figure 3.12). The improvement was significant on four vacuum and one appliance categories. At the top level, the human-generated context and bigrams gave errors of 7.31% and 7.02%, whereas intersecting the two decreased the error to 6.25%. The improvement was significant on one vacuum and three appliance categories (Table 3.10). Therefore, both the human-generated context and the bigrams had constant error rates. This means that bigrams also helped the supervised method scale without harming performance. However, in both the specific and the general setups, intersecting the two types of context proved to be more efficient. This

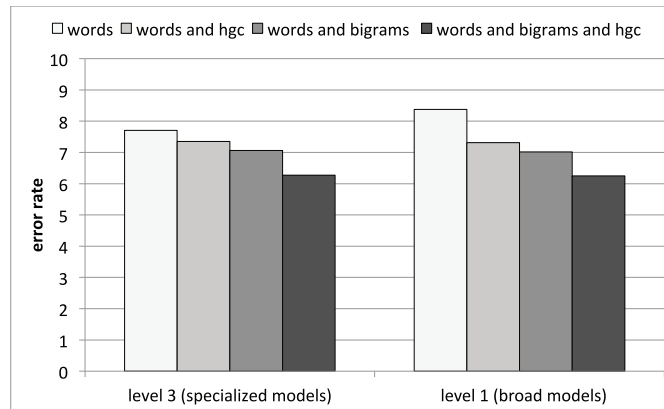


Figure 3.12: Context acquisition. Error of supervised models with the feature space extension. First, word models extended with hgc. Then, extension of word and bigram models

reduced feature space further decreased the error and still helped the method scale on the broad domain. Human-generated context thus improved the method even if bigrams were already present.

3.5 Conclusions

Methods for sentiment classification mostly rely on the polarities of individual words. This harms their performance on broad domains, where sentiment words often have conflicting polarities in different contexts. Supervised methods have been used to learn the polarities of longer word combinations, but with mixed results. Nonetheless, even by analyzing a single sentence, humans are capable of correctly identifying the words that contribute to its sentiment, as well as the contexts that influence the polarities of these words. We investigated how context can be acquired using human computation, and how these features can be integrated in sentiment classification methods, making them scale.

We studied the sentiment classification problem on a dataset with vacuum, camera, kitchen appliance, and hotel reviews. We organized these reviews hierarchically, from narrow domains at the bottom, to a broad domain at the top. We studied two well-established sentiment classification methods. We showed that a lexicon method had a low performance on this broad domain. We also showed that a supervised method achieved a reasonable performance through a collection of specialized classifiers trained at the lower levels of this hierarchy, but that a general model trained at the top degraded performance. We then sought to acquire context features and to integrate them in the lexicon and supervised methods.

We showed that context can be easily acquired at a reasonable price per domain

We designed a human computation task that invited workers to read review sentences and submit answers that contained a sentiment expression, a context, and a polarity. We motivated

Chapter 3. Knowledge Acquisition for Scalable Sentiment Classification

workers through both enjoyment and payment, by packaging the task as a game played for money. We ensured quality through a scoring mechanism that generously rewarded the answers that had common sense and were novel. We recruited workers on a paid crowdsourcing platform and separately acquired context for vacuums, cameras, appliances, and hotels, at a price of roughly \$600-\$750 per domain.

We showed that context helps sentiment classification scale to a broad domain

We explained how context can be used to extend the lexicon and supervised methods. We showed that each separate context model improved these methods on its corresponding domain. We then showed that, by combining these context models, we can reproduce these individual improvements, thus making sentiment classification competitive on a broad domain.

We have thus shown that human computation can deliver a strong performance for the document-level sentiment classification problem.

phrase	context	polarity	phrase	context	polarity
electronics					
small	bin	negative	big	price	negative
small	hose	negative	big	appeal	positive
small	lens	negative	large	button	positive
small	fonts	negative	big	problem	negative
small	price	positive	big	improvement	positive
small	button	negative	big	touch screen	positive
small	display	negative	long	battery life	positive
small	sensor	negative	large	memory card	positive
small	attachment	negative	large	bucket	positive
kitchen appliances					
small	cup	negative	large	jug	positive
small	grinding bin	negative	large	pot	positive
small	oven	negative	large	size	positive
small	toaster	negative	large	capacity	positive
small	capacity	negative	large	ovens	positive
short	cord	negative	large	glass jar	positive
short	lived	negative	long	time	negative
hotels					
small	tv's	negative	big	tub	positive
small	workout room	negative	big	pool	positive
small	rooms	negative	big	room	positive
small	bottle of shampoo	negative	big	charges	negative
small	mirror	negative	big	bathroom	positive
small	drawers	negative	big	breakfast	positive
short	staff	negative	big	balcony	positive
short	duration	negative	big	complaints	negative

Table 3.3: Context acquisition. Sample phrase and context pairs obtained with human computation

Chapter 3. Knowledge Acquisition for Scalable Sentiment Classification

category	lexicon	level 3		level 1	
		+hgc	p-value	+hgc	p-value
vacuum cleaner categories					
hand held	32.89	17.97	2.8E-23	14.07	3.0E-34
canister	29.74	13.20	3.0E-29	10.87	3.4E-37
stand up	30.69	16.91	1.9E-64	13.78	5.6E-93
robotic	39.20	22.83	9.2E-18	18.15	6.9E-29
digital camera categories					
point and shoot	30.73	13.57	2E-169	11.44	2E-216
video	31.61	16.13	2.7E-58	13.16	5.1E-86
kitchen appliance categories					
blender	31.65	15.06	1.1E-71	13.57	4.1E-90
coffee machine	35.41	16.89	1E-205	15.42	4E-251
oven	34.83	14.61	8.0E-88	12.70	6E-110
grill	35.99	15.52	4.9E-41	14.26	1.3E-46
hotel	28.59	12.43	4E-101	11.46	4E-105

Table 3.4: Context acquisition. Error of the lexicon with the sentiment model extension. Left, Hu and Liu [40] lexicon alone. Middle, improvement with the separate hgc at level 3. Right, improvement with the combined hgc at level 1

category	level 3			level 1		
	hgc-iw	hgc-all	p-value	hgc-iw	hgc-all	p-value
vacuum cleaner categories						
hand held	31.27	17.97	8.3E-22	30.23	14.07	4.0E-27
canister	26.30	13.20	3.3E-23	25.37	10.87	9.8E-25
stand up	27.97	16.91	1.6E-50	27.37	13.78	1.6E-66
robotic	38.42	22.83	1.9E-18	34.30	18.15	3.1E-19
digital camera categories						
point and shoot	26.86	13.57	2E-124	26.81	11.44	8E-162
video	27.96	16.13	6.3E-43	28.15	13.16	1.2E-66
kitchen appliance categories						
blender	27.81	15.06	9.5E-54	27.78	13.57	7.2E-63
coffee machine	29.65	16.89	2E-129	31.87	15.42	5E-195
oven	29.16	14.61	1.6E-56	30.15	12.70	1.4E-80
grill	31.46	15.52	8.7E-30	33.47	14.26	1.4E-40
hotel	23.94	12.43	6.4E-66	25.26	11.46	2.7E-81

Table 3.5: Context acquisition. Error of the lexicon with the sentiment model extension. Left, Hu and Liu lexicon extended with the individual words in the context at level 3 (hgc-iw), then with all the context at level 3 (hgc-all). Right, extended with hgc at level 1

category	level 1		
	svm	+hgc	p-value
vacuum cleaner categories			
hand held	7.79	6.84	1.0E-1
canister	5.76	5.48	5.6E-1
stand up	6.51	6.45	8.5E-1
robotic	10.02	9.69	5.6E-1
digital camera categories			
point and shoot	6.12	5.76	4.5E-2
video	6.56	6.43	6.7E-1
appliance categories			
blender	8.43	8.30	6.8E-1
coffee machine	9.18	8.67	1.4E-2
oven	7.68	7.39	3.6E-1
grill	9.06	8.14	9.3E-2
hotel	11.06	9.65	2.2E-6

Table 3.6: Context acquisition. Error of the supervised method with the sentiment score extension. Left, individual words model at level 1. Right, extended with hgc at level 1

category	level 1		
	hgc-iw	hgc-all	p-value
vacuum cleaner categories			
hand held	7.79	6.84	1.1E-1
canister	6.23	5.48	1.7E-1
stand up	6.70	6.45	4.4E-1
robotic	10.02	9.69	6.1E-1
digital camera categories			
point and shoot	6.47	5.76	4.4E-4
video	7.15	6.43	2.3E-2
appliance categories			
blender	9.30	8.30	2.5E-3
coffee machine	9.87	8.67	1.3E-8
oven	7.98	7.39	7.8E-2
grill	9.48	8.14	6.0E-3
hotel	11.03	9.65	4.1E-6

Table 3.7: Context acquisition. Error of the supervised method with the sentiment score extension. Left, individual words model at level 1 extended with hgc-iw at level 1. Right, extended with hgc-all at level 1

Chapter 3. Knowledge Acquisition for Scalable Sentiment Classification

category	level 3			level 1		
	svm	+hgc	p-value	svm	+hgc	p-value
vacuum cleaners						
hand held	7.79	7.03	3.1E-1	7.79	5.32	9.4E-4
canister	6.13	6.04	9.0E-1	5.76	5.76	1.0E-0
stand up	6.54	5.94	1.0E-1	6.51	5.50	8.4E-3
robotic	10.02	9.13	2.8E-1	10.02	8.91	2.3E-1
digital cameras						
point and shoot	5.74	5.61	6.1E-1	6.12	5.99	6.3E-1
video	6.60	6.24	3.9E-1	6.56	6.47	8.3E-1
kitchen appliances						
blender	7.48	7.22	5.3E-1	8.43	7.06	2.2E-3
coffee machine	8.73	8.09	1.1E-2	9.18	7.74	3.6E-7
oven	7.35	6.56	4.9E-2	7.68	6.20	5.3E-4
grill	8.89	9.48	3.9E-1	9.06	7.97	1.3E-1
hotel	8.93	8.60	3.4E-1	11.06	9.40	8.4E-5

Table 3.8: Context acquisition. Error of supervised models with the feature space extension. Left, individual words model at level 3, then improvement with hgc at level 3. Right, models at level 1

category	level 3			level 1		
	hgc-iw	hgc-all	p-value	hgc-iw	hgc-all	p-value
vacuum cleaners						
hand held	7.51	7.03	5.2E-1	7.32	5.32	3.2E-3
canister	6.13	6.04	9.0E-1	5.58	5.76	7.7E-1
stand up	6.45	5.94	1.5E-1	6.62	5.50	3.2E-3
robotic	10.24	9.13	1.9E-1	9.80	8.91	3.3E-1
digital cameras						
point and shoot	5.81	5.61	4.3E-1	6.11	5.99	6.7E-1
video	6.53	6.24	4.8E-1	6.83	6.47	4.2E-1
kitchen appliances						
blender	7.58	7.22	3.9E-1	8.23	7.06	7.3E-3
coffee machine	8.74	8.09	9.1E-3	9.12	7.74	6.7E-7
oven	7.42	6.56	3.5E-2	7.95	6.20	5.0E-5
grill	8.64	9.48	2.3E-1	8.39	7.97	5.4E-1
hotel	8.88	8.60	4.2E-1	10.53	9.40	5.6E-3

Table 3.9: Context acquisition. Error of the supervised method with the feature space extension. Left, individual words model at level 3 extended with hgc-iw at level 3, then with hgc-all at level 3. Right, models at level 1

category	level 3			level 1		
	svm	+hgc	p-value	svm	+hgc	p-value
vacuum cleaners						
hand held	8.17	5.23	1.2E-4	6.46	5.32	1.0E-1
canister	6.04	4.00	1.2E-3	4.46	3.72	1.8E-1
stand up	5.72	4.41	3.4E-4	4.68	4.36	3.3E-1
robotic	10.02	7.57	2.2E-3	9.24	6.79	8.9E-4
digital cameras						
point and shoot	5.35	4.93	8.4E-2	4.95	5.01	8.2E-1
video	5.32	5.39	8.7E-1	5.42	5.29	7.5E-1
kitchen appliances						
blender	6.70	6.21	1.9E-1	6.73	5.14	7.3E-5
coffee machine	7.45	6.52	8.6E-5	7.56	5.95	1.3E-10
oven	5.71	5.21	1.8E-1	5.74	4.68	6.4E-3
grill	7.72	6.88	2.0E-1	7.80	6.71	1.2E-1
hotel	8.53	8.43	7.4E-1	9.70	9.18	1.5E-1

Table 3.10: Context acquisition. Error of the corpus method with the feature space extension. Left, words and bigrams model at level 3, then improvement with hgc at level 3. Right, models at level 1

4 Knowledge Acquisition for Generalizable Opinion Extraction

4.1 Introduction

As a second problem that would benefit from commonsense knowledge, we consider fine-grained opinion extraction: the problem of extracting individual opinions and targets from texts. There are two facets to this problem: generating lexicons with opinion and target expressions; and explicitly pinpointing where opinions appear in a text, as well as finding their corresponding targets. We specifically focus on the latter.

Fine-grained opinion extraction knows two main approaches. On the one hand, there are unsupervised learning methods, such as Double Propagation [94]. These generate lexicons of opinions and targets using hand-crafted syntactic rules. Such lexicons can be employed in pinpointing opinions and targets in texts, which can then be paired based on syntax or proximity heuristics. Another option is to pair candidate opinions and targets based on syntax rules derived with supervised learning, which is the other line of work. There are also fully supervised methods that pinpoint the opinions and targets in a text without relying on lexicons. These typically employ machine learning algorithms that classify the tokens in a sentence based on their neighboring terms. Here, Conditional Random Fields are a common choice.

Models for opinion and target extraction are effective on their training domain. However, without explicit measures for transfer learning or domain adaptation [46, 89], they do not perform well on unfamiliar domains. This poses a problem, in that models need to be retrained whenever additional domains need to be handled, which can be time consuming. Instead it would be more convenient to have a model capable of achieving a high performance even on domains it has not been specifically trained on. Such a model could, for instance, be used to extract opinions from test reviews whose domain is not known. However, high-performance general opinion models have so far been out of reach.

One of the keys to this problems lies in the scope of the training corpus. A lot of opinion and target words are specific to particular domains and do not transfer to others. For example, words like *friendly*, *central*, *staff*, and *location* are highly relevant in hotel reviews. However,

they are probably of little use in vacuum cleaner reviews, where people express opinions about the suction power of a vacuum or the quality of its attachments. As a result, a model learning on hotel reviews will not be well-suited for vacuum reviews.

Therefore, one should use a broader corpus with data from multiple domains, which would allow models to become familiar with a more varied opinion and target vocabulary. One could achieve this using an unsupervised method. However, while these approaches work quite well on small datasets, they can extract a lot of errors when applied to big corpora, where they are likely to pick up accidental extractions, which continue to propagate. Moreover, even when applied to a large corpus, these methods might still have only partial coverage on new domains. The alternative is to use supervised approaches. However, these require training data with fine-grained annotations. Obtaining these has so far been expensive, with only a handful of annotators having to cover thousands of sentences. Even when annotations are available for a broader domain, some supervised approaches can still generalize poorly. The partially supervised methods use lexicons that are generated with unsupervised approaches. Therefore, these methods similarly harm performance. Most importantly, they harm recall by not leveraging syntactic features alone when lexicons have no coverage. On the other hand, fully supervised methods are quite susceptible to memorizing word features and do not properly incorporate syntactic cues, even when the training corpus is broad. As a result, they also perform poorly on unfamiliar domains.

Given the limitations of unsupervised approaches, we can more realistically hope to improve generalization through fully supervised methods that better incorporate syntax features. Therefore, in this chapter, we aim to acquire fine-grained annotations at scale using human computation. We acknowledge that humans can use their common sense to pinpoint opinions and targets in a text, even if they are not trained with detailed manuals and paper exercises. For instance, given the sentence *The staff was not friendly, but at least the location was pretty central*, humans can easily find one opinion *not friendly* about the target *staff*, and another one *central* about the target *location*. Our main contributions can be summarized as follows:

- We describe how human computation can be used to acquire fine-grained annotations that have a high accuracy and can be trusted as training data.
- We show how these annotations can be used in a supervised method that effectively leverages both words and syntax to generalize well to other domains.

While this new task is similar to our context acquisition one (Chapter 3), the fundamental difference between the two comes from the fact that a context for a sentiment expression is not always a target, whereas the reverse always stands. For instance, in the sentence above, the negation *not* is a context for the expression *friendly*, but it is not a target. On the contrary, the two terms make up an opinion expressed about the target *staff*. On the other hand, *staff* is a valid context for the word *friendly*, albeit not a very useful one. Therefore, in a sense, the context acquisition task allowed workers to analyze sentences at an even more fine-grained

level. In addition, while in the former we encouraged humans to primarily find contexts for the ambiguous sentiment expressions, here we expect them to find targets even if they do not affect the polarity of the opinions.

Annotation Acquisition

We show how to acquire fine-grained annotations using human computation, taking inspiration from our context acquisition task, similar in nature. We propose a setup in which workers read review sentences and submit annotations that contain: an opinion expression, its target, a polarity label for the opinion, and an entity/aspect disambiguation label for the target. In exchange for their participation, workers are rewarded: with point updates that reflect the quality of their annotations while helping them solve animal puzzles; and with payments that are proportional to their scores. We thus obtain a game played for money, for which we recruit workers on a paid crowdsourcing platform. We use this game to acquire annotations for a broad corpus that contains reviews from seven domains: digital cameras, vacuum cleaners, mattresses, toys, software, hotels, and restaurants. We show that the annotations we collect from workers have a high accuracy and can be trusted as training data.

Annotation Integration

We extract opinion and target pairs with a Support Vector Machine. This model uses syntax and word features to distinguish the syntactic parse tree dependencies that link opinions to targets from those that do not. For each domain, we test several variants of this model: one trained on that domain, one trained on the union of all seven domains, and one trained on the remaining six domains. We compare these model variants with Double Propagation. Our model beats this method by a significant margin, even with its cross-domain variant. We also test this model on the task of extracting the targets of opinions alone, and compare it with a Conditional Random Field. On the individual domains and the union of domains, these two models are comparable. Across domains, our model does not harm performance and substantially outperforms the benchmark. Therefore, human computation helps the supervised models beat Double Propagation. Moreover, our SVM is much stronger across domains, generalizing without harming performance.

The remainder of this chapter is structured as follows. In Sections 4.2 and 4.3, we describe our fine-grained opinion extraction model and our human computation game. In Section 4.4, we present our experiments and results. In Section 4.5, we draw conclusions.

4.2 Fine-grained Opinion Extraction

We aim to assess how human-generated annotations impact the generalization capability of techniques for fine-grained opinion extraction. We consider two different problems:

<p>syntax and subjectivity features:</p> <ul style="list-style-type: none"> • dependency type • part of speech tags of dependent and governor • is the governor also the governor of a negation dependency? (y/n) • is the dependent/governor a sentiment word? (y/n) <p>Does the dependency:</p> <ul style="list-style-type: none"> • link an adjective to a noun? (y/n) • link a verb to a noun? (y/n) • link a subjective word to a noun? (y/n) • contain exactly one adjective? (y/n) • contain exactly one verb? (y/n) • contain exactly one noun? (y/n)
<p>full features also include:</p> <ul style="list-style-type: none"> • the dependent and governor • if the governor is also the governor of a negation, adverbial modifier, or direct object dependency, the dependent of that dependency (in case there are several such dependencies, we use the dependent of the last one, according to the order in which they are listed by the Stanford Parser)

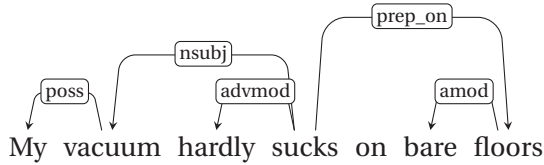
Table 4.1: Annotation acquisition. Features of the Pair SVM for opinion and target extraction

- Extracting both opinions and their targets, a problem that can be solved using lexicons generated with unsupervised methods, like Double Propagation.
- Extracting only the targets of opinions, a problem that can be solved with Conditional Random Fields.

4.2.1 Opinion and Target Pairs

To extract opinions and targets from a sentence, we examine its syntactic parse tree, which we identify using the Stanford Parser. We consider the two words involved in each dependency as a candidate opinion target pair. We first classify if they indeed constitute a positive dependency that links an opinion to a target. If this is the case, we extract the dependent and governor pair, but we do not distinguish which word is the opinion and which is the target. Sometimes the opinion extends beyond the dependency linking it to the target. For instance, the opinion’s semantics can be modified by negations or adverbs (e.g. *camera hardly works*). Moreover, the opinion can contain two separate adjective modifying the same target (e.g. *many interesting dishes*), or can be expressed through the combination of a verb and a direct object (e.g. *camera does wonders*). We thus make an extracted dependency more comprehensive by checking if its governor is also the governor of another dependency that marks a negation, an adjectival modifier, an adverbial modifier, or a direct object. For each such dependency, we append its

dependent word to the extraction. Finally, we sort the extraction’s constituent words by their order of appearance in the sentence.



For example, if, in the sentence above, we classify the dependency marking the verb’s subject *nsubj(sucks, vacuum)* as positive, we extract *vacuum sucks* as an opinion and target pair. We then notice that the governor of this dependency is also involved in another dependency that marks an adverbial modifier *advmod(sucks, hardly)*. We append its dependent to the extraction and obtain *vacuum hardly sucks*.

Support Vector Machine (Pair SVM)

We classify the dependencies in a sentence parse tree as positive (connecting an opinion to a target) or negative using a linear kernel Support Vector Machine. This approach is similar to that of Ku et al. [60]. These authors only classified whether a dependency contained an opinion, based on dependency type rules learned from data. However, we aim to jointly extract opinions and targets. We train a model on a set of positive and negative dependencies, which we represent using two sets of features that capture various properties of each dependency. One set captures syntax and subjectivity characteristics, such as the type of the dependency, the part of speech tags of its constituent words, or whether these words belong to the sentiment lexicon of Hu and Liu [40]. We use another feature set that includes these syntax features as well as the constituent words of that dependency (Table 4.1).

We train these models using the SMO classifier implementation in Weka, by fixing the complexity constant C to 0.1. In our initial experiments, we noticed that the method can output some false positives that link opinions to non-targets or targets to non-opinions. We fix this by increasing the threshold for the Pair SVM model to something higher than zero. We use thresholds of 0.2 and 0.1 for the syntax and full features, respectively. We then sort the remaining positive dependencies by their classification scores. We keep only a certain number of the highest ranked ones, while making sure to include all the dependencies that are tied for the lowest ranked position considered. We keep the top two dependencies.

Double Propagation (DP)

We compare the Pair SVM with an unsupervised benchmark. We decide that a dependency is positive if it links an adjective or verb from an opinion lexicon to a noun from a target lexicon. We build these lexicons using Double Propagation. The original method started from a few seed opinion words and iteratively grew the two lexicons. Instead, we achieve this starting

Chapter 4. Knowledge Acquisition for Generalizable Opinion Extraction

from a few target words. Each iteration has four steps. Two steps extract targets from known opinion words and opinion words from known targets. The original method achieved this by exploring dependencies that linked nouns to adjectives. We use similar dependencies, adapted for the Stanford Parser (*nsubj*, *amod*, *nsubjpass*, *rcmod*), and add new ones that connect nouns to verbs (*iobj*, *dobj*, *rcmod*). We also include some preposition dependencies that connect nouns to either adjectives or verbs (*prep_for*, *prep_at*, *prep_in*, *prep_to*). The other two steps extract opinion words from known opinions and target words from known targets. This is achieved through conjunction dependencies (*conj_and*, *conj_or*, *conj_but*).

4.2.2 Targets Alone

The previous approach extracts opinion and target pairs but does not distinguish which words constitute the opinion and which the target. We also aim to explicitly identify the targets.

Support Vector Machine (Target SVM)

We derive our approach from the pair extraction method. We use the Pair SVM model to classify dependencies as positive or negative. For this task, we drop the measures that we take to reduce the false positives. We keep not at most two, but all positive dependencies. For each such dependency, we check whether its dependent is a candidate target: a non-subjective noun or personal pronoun. If so, we extract the dependent as a target. We then do the same for the governor of a positive dependency.

Conditional Random Field (CRF)

We compare our approach with a linear chain Conditional Random Field. We take inspiration from Jakob and Gurevych [45], who considered the opinion annotations already given and extracted their targets. As features for each token, they used the words and their part of speech tags. Additionally, they used the opinion annotations to compute features that marked: the tokens with which these were involved in direct dependencies, the nouns to which these were closest in a sentence, and all the tokens with which these co-occurred in a sentence. The model classified tokens as: the beginning or continuation of a target, or something else. However, we consider opinion annotations are unknown beforehand, so we cannot use them to compute exactly the same features. Moreover, through our task design, we only acquire training sentences that contain at least one opinion annotation. Finally, we evaluate models not on exact matches, but based on overlap. With this evaluation, classifying tokens as beginning of continuation of targets can sometimes give long extractions that are unfairly considered correct.

We thus use the following model. We represent each word with two feature sets. One set captures syntax and subjectivity. For each word, this includes its part of speech tag. Moreover, if the word is in a direct dependency with a sentiment word from the Hu and Liu lexicon, we

output a direct dependency marker feature. We also output another feature that marks the type of that dependency, since this information is generally useful to discriminate targets from other words. In addition, we mark the nouns closest to subjective words with a word distance feature. Another feature set includes the syntax features and the words themselves. Therefore, we compute the direct dependency and word distance features based on a sentiment lexicon and not based on the opinion annotations. Moreover, we only train on sentences that have opinion annotations, so there is no need for features that mark the tokens in opinionated sentences. Finally, to avoid the issue of long extractions, this model classifies words as targets or something else, and we evaluate the correctness of each extracted word. We train the model using the SimpleTagger implementation in MALLET [79].

4.3 Human Computation Task

We acquire fine-grained opinion annotations using human computation. When designing this new task, we keep the same considerations in mind as for the context acquisition task:

- We need to obtain information in a structured way while still allowing workers to express complex knowledge.
- We need to recruit workers and motivate them to invest effort in the task.
- We need to ensure that workers understand the task and are qualified to do it.
- We need to control the quality of answers.

4.3.1 Task Structure

To obtain information in a focussed way, we structure our task in rounds, where in each round a worker needs to annotate a sentence extracted from an review. For each sentence, we require workers to express complex knowledge, by constructing an annotation in four steps:

1. *Opinion*: highlighting an opinion expression within the sentence.
2. *Target*: highlighting a target for the identified opinion.
3. *Polarity*: choosing a positive or negative polarity for the opinion.
4. *Entity/Aspect*: choosing if the target is an entity or an aspect.

When a worker submits an annotation, she starts a new round. She can also skip a round, if the sentence does not contain an opinion, or she is unsure how to annotate.

Worker Recruitment and Motivation

To increase motivation and encourage workers to stay with the task for a longer period of time, we rely on enjoyment and payment. We reward workers with: score updates that allow them to gradually solve an animal puzzle assigned to them (we reuse the puzzles from the context acquisition task); and with small payments that are proportional to their scores. We again obtain a game played for money (Figure 4.1), for which we recruit workers on Amazon Mechanical Turk.

Sandbox Stage

To ensure that workers understand the task and are qualified to do it, one option would be to create an interactive tutorial, similar to what we used in the context acquisition game. However, we decide to replace this tutorial with a sandbox stage at the start of the game. This stage trains the workers but at the same time allows them to directly solve the real task. It consists of nine rounds with simple sentences (e.g. *The hallway was noisy*). Each sandbox sentence has gold-standard annotations attached (e.g. in the previous sentence: the opinion *noisy* is expressed about the target *hallway*; the opinion is negative; and the target is an aspect). During the sandbox rounds, we guide workers with short instructions embedded in the interface, explaining how the four annotation components should be constructed (Figure 4.2). As workers fill in the various annotation components, the corresponding instruction boxes switch color from dark grey to light green, thus indicating to workers that they are on the right track. If workers submit correct annotations in at least seven of the nine rounds, they graduate the sandbox and move on to annotate real sentences. We consider an annotation is correct if it agrees with one of our annotations on all four components (two text selections agree if they overlap).

4.3.2 Quality Assurance

The sandbox stage implicitly influences quality before the game, by allowing only the workers that have gained a good understanding to proceed to solve the real task. The scoring mechanism controls quality during the game, by encouraging workers to submit good quality annotations. Similar to the context acquisition game, we again lock workers after 200 answers or, if after at least fifty answers, their average score falls below a predefined threshold. After the game, we control quality by first filtering out the bad workers, then by aggregating the remaining annotations.

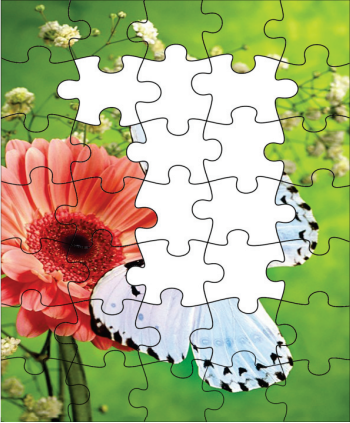
Scoring Mechanism

During the game, we use a scoring mechanism that rewards good annotations. For the nine sandbox rounds, we use a simple scoring function that compares a new annotation with the gold-standard ones, and awards twenty points if we can find a gold annotation with which

4.3. Human Computation Task

0 points

Welcome **marina.boia!**
[Check out \(\\$0.30\)](#) [I need help](#) [Sign out](#)



These animals taste with their feet.

It is a friendly place with a wonderful staff.

Unsure how to construct the answer. Skip to the next sentence!

1. **OPINION** *

➔

Clear
2. **TARGET** *

➔

Clear
3. **IS THE OPINION?** *

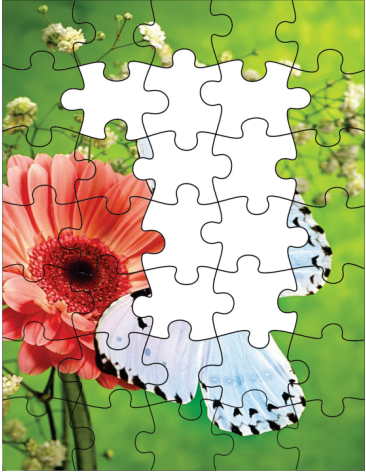
positive
 negative
4. **IS THE TARGET?** *

entity
 feature
 unsure

Submit

Figure 4.1: Annotation acquisition. Main game interface

This is a puzzle about an animal. Submit answers and find out what it's about.



These animals taste with their feet.

This is a sentence from a hotel review.

From this sentence, construct an answer with: 1) an opinion, 2) a target, 3) if the opinion is positive or negative, 4) if the target is an entity or feature.

If the sentence contains multiple opinions, just choose one, then find its target.

The hotel was beautiful, but it had horrible customer service.

1. **OPINION** *

An opinion phrase is a word or word combination that is positive or negative about something (e.g. the hotel is nice). Highlight such a phrase in the text, then click on the purple arrow.

➔

Clear
2. **TARGET** *

A target phrase is a word or word combination that shows what the opinion is about (e.g. great location). Highlight such a phrase in the text, then click on the orange arrow.

➔

Clear
3. **IS THE OPINION?** *

Indicate whether the opinion you identified is positive or negative.

positive
 negative
4. **IS THE TARGET?** *

Indicate whether the target is an **entity** (e.g. a hotel) or a **feature** (e.g. its location, staff, rooms, hallway, pool, price).

entity
 feature
 unsure

Figure 4.2: Annotation acquisition. Example of a training round in the sandbox stage

Chapter 4. Knowledge Acquisition for Generalizable Opinion Extraction

the new one is in full agreement, on all four components. For the regular rounds, we score a new annotation highest if it agrees with those submitted by previous workers. More precisely, we compare it with the annotations previously acquired for that same sentence. For each prior annotation, we determine on how many components it agrees with the current one. We compute a tentative score that sums: ten, fifteen, five, and five points in case of opinion, target, polarity, and entity/aspect agreement, respectively. We heuristically established these amounts based on the effort required in identifying the four annotation components (from the workers' behavior in initial experiments, we observed that identifying the target of the opinion is a more complex step than identifying the opinion itself; additionally the opinion and target selection steps are more difficult than the steps requiring workers to find the polarity of the opinion or to specify whether the target is an entity or an aspect). For instance, a prior annotation that agrees with the current one on the opinion and target will give a tentative score of twenty-five points. We finally loop over all prior annotations and return the maximum tentative score. When there are no prior annotations, we revert to the original scoring function used in the context acquisition game.

Worker Filtering

After the game, we remove the workers that have a bad performance on gold-standard game rounds. We define the gold rounds by selecting forty sentences of varied complexity, to which we attach all possible annotations. We interleave these gold rounds with the regular rounds in the game. We consider that a worker's activity is satisfactory if she submits correct annotations in the majority of the gold rounds assigned to her. Here, we consider an annotation is correct if it agrees with one of our own on both the opinion and the target components.

Answer Aggregation

Finally, we iterate over the game sentences and aggregate the acquired annotations. For each sentence, we group the annotations that capture the same opinion and target. For each annotation, we consider the existing groups one a time. If the current annotation overlaps in both the opinion and the target with all the annotations in an existing group, we place it in that group. If we find no such group, we create a new one with that annotation. After constructing the groups, we collapse each one into an aggregate annotation. To aggregate the opinion and target selections, we take their most frequent start and end boundaries among all the annotations in the group. However, when there are ties, we choose the inner most boundaries. To the resulting opinion and target selections, we attach the majority polarity and entity/aspect labels. However, if there are ties, we drop that aggregate annotation. Finally, we consider only those groups which contain a number of annotations above a predefined threshold. For the gold sentences, which are seen by many workers, we set this threshold to seven. For the regular sentences, which are seen by at most five players, we set the threshold to two (Table 4.2 shows an example).

Fast check-in and the view is great!			
opinion	target	polarity	entity/aspect
great	view	positive	aspect
great	view	positive	aspect
great	the view	negative	entity
great	view	positive	aspect
fast	check	negative	aspect
fast	check-in	positive	entity
fast	check-in	positive	aspect
fast	check-in	positive	aspect
great	check-in	positive	aspect
-	-	-	-

Table 4.2: Annotation acquisition. Example of a game round and individual annotations obtained, along with the final aggregate annotations (in bold)

4.4 Empirical Results

We tested the human computation design using review data from several domains. We launched the game seven times using sentences from camera, vacuum, mattress, toys, software, hotel, and restaurant domains, respectively. We recruited workers from Amazon Mechanical Turk. After quality assurance, we ended up with 7,700 annotations for 6,900 sentences. We compared our methods with a Conditional Random Field and with Double Propagation.

We present results about the performance of the techniques we developed, structured around the major conclusions that they support.

4.4.1 Dataset

We used approximately 4,000 camera reviews, 4,000 vacuum reviews, 5,850 mattress reviews, 1,550 reviews of toys, 5,000 software reviews (mostly video games), 4,000 hotel reviews, and 4,000 restaurant reviews. We ran our task for each of these domains. The camera, vacuum, and hotel reviews came from the Amazon and Tripadvisor datasets described in Chapter 3. Part of the mattress reviews came from the corpus used by Zhang and Liu [136], which we complemented with other reviews downloaded from Amazon. The toy and restaurant reviews came from Epinions. Finally, the software reviews were extracted from the corpus of Jindal and Liu [48].

4.4.2 Task Setup

To set up the task for one domain, we first instantiated the game rounds with review sentences. We did not want workers to become frustrated from having to skip a lot of rounds that displayed overly complex or non-opinionated sentences, so we added a few constraints. We ensured that

the sentences were no longer than twenty tokens. Moreover, we ensured that they were likely to contain opinions. We used the union of three sentiment lexicons (General Inquirer [110], OpinionFinder [130], and the lexicon of Hu and Liu) to find the sentences that contained at least one and no more than three subjective words. In addition, we ran Double Propagation on the review corpus associated to that domain and obtained the resulting opinion and target lexicons. We then restricted to the sentences that contained at least one of the most frequent fifty to eighty targets, either modified by a subjective word, an adjective, or an adverb. Note that we did not consider sentences containing comparative adjectives or adverbs. These are more complex to annotate, given that they require the identification of an opinion, along with the two or more items that this compares. We constructed 1,500 rounds with such sentences. We browsed several game rounds, marking forty of them as gold standard. We also created nine simple sentences for the sandbox stage. We attached possible annotations to both subsets.

We then launched the game and collected annotations. We recruited workers through Amazon Mechanical Turk. We rewarded each worker with a base payment of \$0.3, which they earned when completing the sandbox, and with a bonus of \$0.06 for every 100 points that they earned in the game after that. We filtered workers based on their performance on the gold rounds, then aggregated the remaining annotations. We observed that, for most domains, roughly 250 workers suffice to obtain at least one aggregate annotation for approximately 1,000 sentences.

Finally, we browsed multiple sentences that were not part of the gold or sandbox rounds, for which we had obtained at least one aggregate annotation. We selected between 160 and 350 such sentences, to which we also attached possible annotations. We used all of these gold annotations to evaluate the quality of the workers' aggregate annotations.

4.4.3 Annotation Evaluation

Human computation produces fine-grained annotations with high accuracy

Following the above steps, we created a resource with 6,900 review sentences for which we have 7,700 opinion and target annotations. For each domain, we have roughly 1,000 sentences, each with at least one aggregate annotation. We assessed the quality of these annotations using our gold standard, consisting of a subset of 1,700 sentences, for which we have 2,300 annotations (Table 4.3).

For each aggregate annotation attached to a sentence in our gold standard, we checked if it agreed with one of our own annotations. Conversely, for each of our annotations, we checked whether it agreed with one of the aggregate annotations. We thus computed precision and recall measures that captured the correctness and coverage of the aggregate annotations. We computed these measures using four different agreement measures. We recorded f-scores for each domain, then the average performance.

We first considered two annotations agreed if they overlapped in their opinion components,

domain	gold annotations		worker annotations	
	sentences	pairs	sentences	pairs
camera	393	481	1,269	1,433
vacuum	219	310	945	1,061
mattress	222	293	1,002	1,104
toy	209	284	838	930
software	224	318	994	1,102
hotel	219	316	924	1,046
restaurant	219	300	921	1,037
total	1,705	2,302	6,893	7,713

Table 4.3: Annotation acquisition. Number of sentences and opinion target pairs in the gold annotations and in workers’ aggregate annotations

domain	worker annotation f-scores			
	opinion	target	opn-trg	full
camera	0.946	0.899	0.892	0.883
vacuum	0.898	0.823	0.793	0.783
mattress	0.943	0.866	0.879	0.848
toy	0.905	0.788	0.779	0.735
software	0.895	0.843	0.851	0.821
hotel	0.893	0.815	0.793	0.779
restaurant	0.903	0.821	0.806	0.792
average	0.912	0.836	0.828	0.806

Table 4.4: Annotation acquisition. First, annotation f-scores based on agreement on the opinion and target components, respectively. Next, joint agreement on both the opinion and the target (opn-trg). Finally, full agreement on all annotations components (full)

then looked at the agreement based on target overlap. Workers identified the opinions and targets with average f-scores of 0.912 and 0.836, respectively (Table 4.4). Thus, workers learned to almost perfectly identify the opinions. They also identified the targets with a high accuracy. This was lower than the opinion f-score, partly because some opinions hierarchically apply to several targets in a sentence. For instance, in a sentence like *This camera is a good size*, we were inclined to annotate the opinion *good* about the target *size*. However, workers were split between this annotation and the higher-level alternative in which the opinion *good size* is about the *camera*. In a third agreement measure, we required overlap in both the opinion and the target, which gave an average f-score of 0.828. Finally, we added more constraints and required agreement in all four annotation components, including the polarity and entity/aspect components. This gave an average f-score of 0.806. Therefore, workers were not only able to identify opinions and targets, but also managed to reliably pair them and to further describe them in terms of polarity and the target being an entity or an aspect.

This proves that the annotations we obtained with human computation are of high quality and can be trusted as training data. Moreover, the high agreement with the workers’ annotations

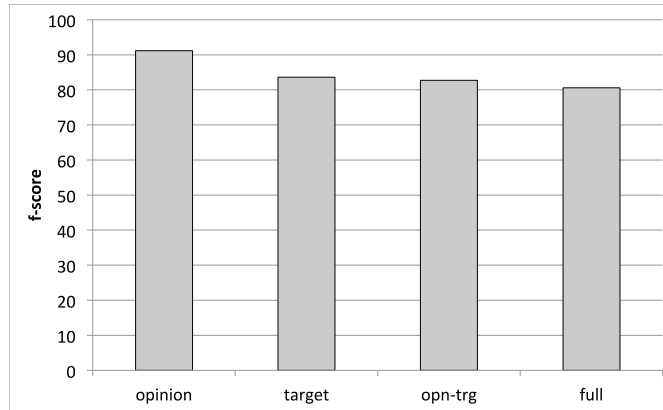


Figure 4.3: Annotation acquisition. Average annotation f-scores

domain	dp	pair svm					
		syntax features			full features		
		id	ud	cd	id	ud	cd
camera	0.588	0.732	0.739	0.705	0.748	0.746	0.738
vacuum	0.497	0.617	0.624	0.599	0.645	0.669	0.629
mattress	0.553	0.640	0.667	0.671	0.652	0.674	0.669
toy	0.414	0.496	0.559	0.554	0.511	0.591	0.602
software	0.506	0.593	0.618	0.604	0.583	0.641	0.607
hotel	0.540	0.618	0.629	0.605	0.634	0.655	0.644
restaurant	0.488	0.556	0.602	0.598	0.567	0.623	0.616
average	0.512	0.607	0.634	0.619	0.620	0.657	0.644

Table 4.5: Annotation acquisition. Opinion and target extraction f-scores of Double Propagation and the three variants of the Pair SVM model: individual domains (id), union of domains (ud), and cross-domain (cd)

shows our gold annotations can also be trusted.

4.4.4 Model Evaluation

The Pair SVM trained with human-generated annotations beats Double Propagation

We tested the Double Propagation and Pair SVM models for opinion and target extraction. For each domain, we evaluated these models on a test set containing 200 sentences from our gold standard. Given a test sentence, we considered an extraction was correct if it overlapped both the opinion and the target in one of our gold annotations for that same sentence. We recorded f-scores for each domain, then the average performance.

We ran Double Propagation on the review set corresponding to each domain and obtained opinion and target lexicons. We used these lexicons to extract opinions and targets. We then trained the Pair SVM model to classify parse tree dependencies. Its training set included all

the sentences that were not used for testing and for which we had aggregate annotations. Given a training sentence, we defined a training instance for each dependency in the sentence parse tree. We represented these dependencies using first the syntax and then the full feature sets. We considered as positive the dependencies whose two constituent words overlapped both the opinion and the target in one of the aggregate annotations for that sentence. We attached negative labels to all the other dependencies. To fix the class imbalance, we randomly dropped some negative examples. We used the resulting Pair SVM to extract opinions and targets. For each domain, we evaluated three variants of this model: an individual domain variant obtained on the training data for that domain; a union domain variant trained on the data from all seven domains; and a cross-domain variant trained on the data from the remaining six domains.

Double Propagation gave an average f-score of 0.512 (Table 4.5). The individual domain variant of the Pair SVM model gave an average f-score of 0.607 when using the syntax features. Therefore, the Pair SVM model learned better syntax rules that outperformed Double Propagation. For instance, the Pair SVM often assigned a high weight to the dependency type *nsubj* (e.g. *camera is good*) and a smaller or even null weight to the dependency type *amod* (e.g. *good camera*). Both of these dependency types are used in Double Propagation. This shows that *amod* dependencies can also extract a lot of non-opinions (e.g. *digital camera, optical zoom*), and that opinions are more likely expressed through the *nsubj* pattern. Moreover, the better performance is probably also due to the Pair SVM relying on several other syntax features that are not captured by the Double Propagation heuristics. When using the full feature set, the individual domain variant of the Pair SVM gave an average f-score of 0.620. This variant learned similar syntax features. In addition, it also assigned positive weights to subjective, entity, and aspect words. It thus incorporated knowledge about which words make opinion and target pairs. This helped improve performance a bit further.

The union domain variant of the Pair SVM model gave average f-scores of 0.634 and 0.657 when using the syntax and the full feature sets, respectively. The performance was better than that of the individual domain variant, with a slightly more noticeable improvement recorded when using the full feature set. Because these models were trained using roughly seven times more training data, it might be that they were able to infer syntax rules that worked better on some domains. Moreover, the increase in training data enabled this model to also incorporate better knowledge about which words constitute opinion and target pairs.

The cross-domain variant of the Pair SVM model gave average f-scores of 0.619 and 0.644 when using the syntax and the full feature sets, respectively. Therefore, this model also performed slightly better than the individual domain variant. This means it also benefitted from more training data and learned more effective syntax rules. Moreover, when this model also included word features, it did not drop in performance. This hints that it could still leverage the syntax features, even when opinion and target words did not transfer to the test domain.

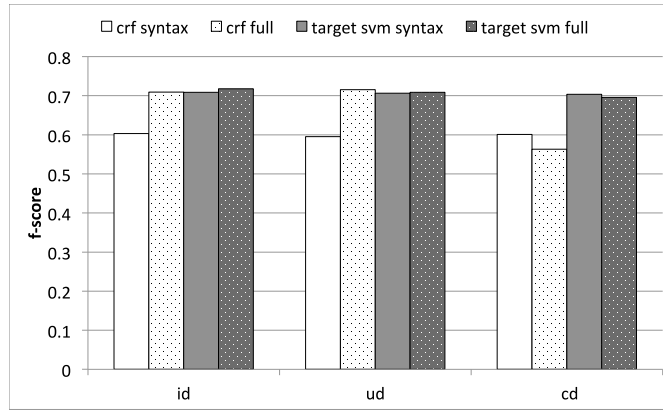


Figure 4.4: Target extraction f-scores for models trained with syntax and full features, in the individual domains (id), union of domains (ud), and cross-domain (cd) setups

The Target SVM is comparable to the CRF when trained on the individual domains and on the union of domains, but outperforms it across domains

We also tested the CRF and Target SVM models for target extraction. We evaluated these models on the same test sets that we used for pair extraction. We trained them on the workers' annotations for the remaining sentences. We used both the syntax and full feature sets, in the individual, union, and cross-domain setups. We considered an extraction was correct if it overlapped the target component in one of our gold annotations.

When trained on the individual domains, the CRF gave average f-scores of 0.603 and 0.709 with the syntax and full feature sets, respectively (Figure 4.4). When trained on the union of domains, its f-scores were in the same vicinity. However, across domains, its performance decreased to 0.563 when using the full features. Therefore, the model had a relatively constant performance when using the syntax features, even across domains. Moreover, the model substantially improved when it complemented the syntax features with the actual words. However, this happened only when it was trained on the individual domains and on the union of the seven domains, when the opinion and target words encountered during training most probably also transferred to the test sets. On the other hand, when tested across domains, some of the learned opinion and target words did not transfer. This harmed the model's performance, which was below that obtained when using the syntax features alone. This hints that, when the model is trained on both syntax and word features, it assigns most of the weight to the latter (Jakob and Gurevych [45] noticed a similar behavior). As a result, in individual and union domain setups, the model can exploit the word features to achieve a good performance. However, across domains, the model has to rely on the poorly learned syntax features and thus harms performance.

In comparison, on the individual domains, the Target SVM model gave average f-scores of 0.709 and 0.717 with the syntax and full features, respectively. On the union of domains, the model's performance was in the same vicinity. Across domains, the model only slightly

domain	syntax features		full features	
	crf	target svm	crf	target svm
camera	0.764	0.783	0.819	0.794
vacuum	0.563	0.689	0.687	0.712
mattress	0.691	0.694	0.737	0.713
toy	0.455	0.595	0.603	0.614
software	0.629	0.708	0.684	0.688
hotel	0.576	0.761	0.727	0.766
restaurant	0.544	0.730	0.709	0.735
average	0.603	0.709	0.709	0.717

Table 4.6: Annotation acquisition. Target extraction f-scores for models trained on the individual domains

domain	syntax features		full features	
	crf	target svm	crf	target svm
camera	0.731	0.787	0.823	0.768
vacuum	0.525	0.680	0.700	0.718
mattress	0.630	0.710	0.724	0.687
toy	0.509	0.647	0.629	0.650
software	0.587	0.696	0.717	0.682
hotel	0.608	0.740	0.744	0.742
restaurant	0.578	0.687	0.670	0.713
average	0.596	0.707	0.715	0.709

Table 4.7: Annotation acquisition. Target extraction f-scores for models trained on the union of domains

domain	syntax features		full features	
	crf	target svm	crf	target svm
camera	0.719	0.781	0.710	0.772
vacuum	0.515	0.701	0.461	0.690
mattress	0.624	0.714	0.664	0.662
toy	0.552	0.650	0.518	0.670
software	0.582	0.687	0.497	0.650
hotel	0.641	0.713	0.575	0.737
restaurant	0.575	0.681	0.516	0.688
average	0.601	0.704	0.563	0.696

Table 4.8: Annotation acquisition. Target extraction f-scores for models tested across domains

decreased in performance when using the full feature set. Therefore, when using the syntax features alone, the Target SVM consistently outperformed the CRF, with all three training variants. This hints that the syntax features used by this model are more expressive. The CRF considers words in a sequence, so it captures features only individually for each word. Its syntax features do not go beyond the part of speech tags, the short dependency, and word distance markers. To label a word, the CRF does use the label of the neighboring tokens, but this is not very useful, given that the model does not extract opinions, so adjacency to opinion labels is not exploited. In contrast, the Target SVM considers dependency word pairs. It exploits more expressive syntactic information about both words in a pair, like their part of speech tags, the dependency type, or boolean markers capturing whether the pair links a noun to an adjective, verb, or subjective word. The model also includes a feature marking whether the governor of a dependency is negated. Such predictors are not captured by the CRF, which is probably why this model has a lower performance. When using the full feature sets, the two models had a similar performance in the individual domain and union of domains setups. However, across domains, the Target SVM was more resilient to word features not transferring, probably because it does not assign negligible weights to syntax features once words are added. It thus maintained a roughly constant performance and beat the CRF.

4.5 Conclusions

Methods for fine-grained opinion extractions perform reasonably well on their training domain but cannot generalize to new ones. For models to improve their generalization capabilities, they need to be trained on a broader corpus with data from multiple domains. However, this leads state-of-the-art unsupervised methods like Double Propagation to pick up more errors. For supervised methods, this is a problem because fine-grained annotations are expensive to obtain at scale. In addition, state-of-the-art approaches like Conditional Random Fields heavily rely on word features and thus produce errors on new domains, where these models cannot effectively leverage syntactic cues alone, once word features do not transfer.

We showed that fine-grained annotations can be acquired with high accuracy

We showed how to acquire opinion and target annotations using human computation. We proposed a paid game in which workers read review sentences and submitted annotations with four components: an opinion, its target, a polarity label for the opinion, and an entity/aspect disambiguation label for the target. In exchange, workers were rewarded with point updates that reflected the quality of their annotations and with payments proportional to their scores. By launching this game on a crowdsourcing platform, we acquired annotations for a broad corpus containing seven domains: cameras, vacuums, mattresses, toys, software, hotels, and restaurants. We showed that workers submitted annotations of good quality, which means that these can be trusted as training data for fine-grained opinion models.

sentence with opinion and target selections	polarity	entity / aspect
digital cameras		
<i>Overall, I can say that this is one of the best product I've bought in 2010.</i>	positive	entity
<i>You even get sound when you do under water video!</i>	positive	aspect
<i>However when I got my Z1015 from Amazon and started to use it, it was really annoying.</i>	negative	entity
<i>The price is exceptionally reasonable considering all you get.</i>	positive	aspect
mattresses		
<i>I worried about the smell after so many reviews mentioned it.</i>	negative	aspect
<i>The springs in this mattress do it no favors, I feel every one of them.</i>	negative	aspect
<i>The springs in this mattress do it no favors, I feel every one of them.</i>	negative	aspect
<i>After putting some nice sheets on, the bed was very comfortable.</i>	positive	entity
restaurants		
<i>The dining area is small, and full of uncomfortable tables.</i>	negative	aspect
<i>The dining area is small, and full of uncomfortable tables.</i>	negative	aspect
<i>My steak wasn't the only thing that was wrong.</i>	negative	aspect
<i>It is truly a quaint and charming place.</i>	positive	entity

Table 4.9: Annotation acquisition. Sample annotations obtained for camera, mattress, and restaurant reviews

We showed that the proposed SVM significantly beat Double Propagation and generalized much better than the CRF

We then proposed an SVM that extracted opinions and targets by classifying parse tree dependencies using syntax and word features. We trained this model on the annotations we acquired using human computation. For each domain, we tested three variants of this model: one trained on the individual domains, one trained on the union of domains, and a leave-one-out cross-domain variant trained on the remaining six domains. Either with individual domain, union of domains, or cross-domain variants, our model always outperformed the unsupervised benchmark - Double Propagation. We also compared this model with a supervised benchmark - a CRF, which we also trained on the annotations acquired with human computation. With the individual domain and union of domain variants, the two models had a comparable performance. However, across domains, our model significantly outperformed the benchmark. Therefore, models trained with human-generated annotations significantly beat Double Propagation. Moreover, unlike the CRF, our model showed it can generalize to new domains without harming performance.

We have thus shown that human computation can help deliver a strong performance for the fine-grained opinion extraction problem.

5 Sentiment Knowledge Acquisition with Volunteers

5.1 Introduction

So far in this thesis, we have studied two sentiment analysis problems: document-level sentiment classification and fine-grained opinion extraction. We investigated how commonsense knowledge acquired through human computation can improve the performance of automatic approaches designed to solve these problems. With a first task, we acquired knowledge about the polarities of sentiment words in various contexts, which we used to improve the performance of models for sentiment classification. With a second task, we asked workers to analyze opinionated sentences and pinpoint individual opinion expressions along with their corresponding targets. We used this knowledge to obtain an improved model for fine-grained opinion extraction. To acquire such commonsense knowledge, we designed two human computation games, for which we recruited paid workers on the Amazon Mechanical Turk crowdsourcing platform. Therefore, up to this point, we have been motivating workers through both enjoyment and payment, which proved to be a successful recipe for reaching out to a community of workers that were willing to participate and provide good quality knowledge.

In this chapter, we aim to investigate if it is possible to find a task setup that is as effective but does not rely on the payment component to recruit and motivate workers. A key advantage to using voluntary workers is that this would allow to run tasks that collect commonsense knowledge for a wider spectrum of languages. So far, we have researched sentiment analysis performance on texts written in English, the language for which most of the work in this field has been conducted [69]. This worked well, given that the majority of workers on Amazon Mechanical Turk are familiar with the language, most of them coming from the United States or from India [42]. However, on such crowdsourcing platforms, other languages are familiar to a substantially smaller pool of workers. For instance, Mellebeek et al. [81] have tried to recruit Amazon Mechanical Turk workers in order to annotate sentences written in Spanish. However, in their initial trials, results were not encouraging, as most of the participants came from India and solved the task by clicking randomly. The authors thus had to extend their design by introducing a language competence test. While this is an acceptable solution, it

drastically decreases the pool of available workers. As a result, it would be difficult to use our existing task setup to acquire knowledge in other languages. Therefore, this is an important motivation for researching how this can be achieved with the help of voluntary workers.

However, involving volunteers is challenging. First of all, workers need to hear about the task and access it. Secondly, they need to be convinced to participate. There have been a few tasks that succeeded: the ESP game for image labeling [118], Wikipedia, Duolingo, or Zooniverse. However, these successful cases are, in general, difficult to reproduce, given that there is no clear recipe for how a community of engaged volunteers can be built. As a solution to this problem, Ipeirotis and Gabrilovich [43] have illustrated how volunteers can be recruited and engaged through online advertising. They used the Google Adwords platform to run ads that led workers to a task probing their knowledge on specialized topics, such as medicine. In doing so, they relied on the platform's capability to optimize ad placement in order to maximize the number of clicks that led to the desired behavior (workers interacting with the task), also called conversions. The authors further engaged workers by relying on game elements. They thus showed that online advertising and gamification give a viable approach for reaching communities of workers that possess expert knowledge and are willing to share it.

Our goal is to establish whether we can find a recipe for engaging workers to voluntarily participate in human computation tasks for commonsense knowledge acquisition. Specifically, we aim to design tasks that elicit knowledge for sentiment classification: the polarities of individual words and longer word combinations. To highlight the advantage of such an approach, we choose a language other than English, and focus on French. There have been several attempts to analyze opinions in texts written in French [78, 28, 116]. There have also been some attempts that relied on volunteers to acquire sentiment knowledge for the French language. For instance, the games LikeIt and Emot [63] are part of the larger project JeuxDeMots [62]. These collect knowledge about the polarity and emotional charge of phrases sampled from a predefined vocabulary. The JeuxDeMots platform was launched in 2007 and, in time, attracted a large pool of participants. However, it is unclear how this participant base was built. According to initial reports [62], it seems that no special advertising was made and that people learned about the platform through word of mouth. This brings us back to the same problem: it is not evident how to systematically recruit and engage volunteers.

In this chapter, we propose a recipe for recruiting and engaging voluntary workers to provide knowledge for sentiment analysis. First of all, as Ipeirotis and Gabrilovich [43], we use online advertising to attract users to our tasks. Moreover, we make use of the advertising platform's capacity to optimize ad placement in order to reach workers that convert. Second of all, we apply our experience with paid workers, and design our tasks as games that inspire enjoyment. Our main contributions can be summarized as follows:

- We show that online advertising and human computation games give an effective recipe for attracting voluntary workers that are willing to contribute with good quality sentiment knowledge.

- We also compare several game metaphors and hint which one might lead to the optimal conversion rate. In addition, we propose that the choice of game metaphor does not noticeably influence the time spent in the game nor the quality of the contribution, once a worker converts.

Game Metaphor Exploration

We divide our study in two phases. In a first phase, we aim to explore what kind of tasks can convince workers to participate. We choose three different questions that we ask workers: to select individual sentiment words from sentences and indicate their polarities; to label with polarities individual words chosen from a predefined vocabulary; to choose longer word combinations from sentences and indicate their polarities. To motivate workers to participate, we wrap each of these questions around three different game metaphors, respectively: a first one based on uncovering animal puzzles by answering questions, inspired from our previous human computation games; a second one in which a worker is a small explorer that visits villages by avoiding obstacles in a forest, with each visit receiving a new question to answer; a third, similar one, in which the worker is again an explorer, this time needing to pick flowers while navigating through a labyrinth. We thus obtain three games, each one asking workers to solve a different question and wrapped in a different game metaphor. By sequentially running three advertisement campaigns with conversion optimization, we are able to systematically lure workers to these games and convince a reasonable fraction to participate. The workers that convert provide good quality knowledge, with which we manage to improve the sentiment classification performance of an existing sentiment lexicon in French. This shows that, by combining online advertising and games, we obtain a good recipe for luring and convincing voluntary workers to conscientiously solve tasks of varying complexity.

Game Metaphor Comparison

In a second phase, we strive to systematically study which game metaphors are more effective in persuading workers to participate. In doing so, we choose the third, more complex question of selecting longer sentiment features, then wrap it around the three game metaphors proposed: animal puzzles, village explorer, and puzzle explorer. We run these three games in parallel, by advertising them in the same campaign. While our results are not fully conclusive, they do hint that the animal puzzles metaphor might be more effective in motivating workers to take action. However, we also discover that, once workers decide to participate, the choice of game metaphor has no statistically significant impact on how long they stay in the game, nor on the quality of the answers they provide.

In the remainder of this chapter, Section 5.2 describes how we setup our human computation tasks, Section 5.3 presents our results, whereas in Section 5.4 we draw conclusions.

5.2 Recruiting and Engaging Volunteers

We propose to recruit and engage volunteers by relying on two factors. We design our tasks as games, to convince workers to stay with them for a longer time. We advertise the tasks online, to recruit workers that are likely to contribute.

5.2.1 Task Design Exploration

To find a task design that can persuade workers to participate, we conduct an initial exploration phase where we try three different setups. We design tasks that convince volunteers to contribute knowledge for sentiment classification: the polarities of individual words and longer word combinations. This is similar to the context acquisition problem, for which we designed a game that was played by paid workers recruited on Amazon Mechanical Turk (Chapter 3). However, given that we now want to recruit voluntary workers, we need to reconsider our task design. With paid workers, we had a guarantee that these will invest more than one second: to complete the tutorial and thus understand how the game worked; and to play the game for an average of ninety rounds. This is because workers on Amazon Mechanical Turk strive to keep a clean reputation by completing the tasks that they accept. In addition, when we designed the context acquisition game, we coupled the workers' total payment with the number of submitted answers, as well as with the quality of these answers. Therefore, beyond the game metaphor expressed through the scoring mechanism and the animal puzzles, workers had an extrinsic incentive to stay with the task. However, voluntary workers have no track record to worry about and are not paid for their contribution. Therefore, after workers click on our ads, we only have a short time to engage them. We thus need to reconsider the task design so that:

- It is appealing enough to instantly motivate workers to participate for a longer time.
- It is intuitive enough for workers to quickly understand what the task is about and what is required from them.

Additionally, we are still concerned with obtaining knowledge in a structured way and with controlling the quality of answers.

Animal Puzzles

For our first task design, we use the context acquisition game as a source of inspiration. To obtain knowledge in a focussed way, we still structure the task in rounds. However, because this is our initial trial with voluntary workers, we reduce the complexity of the required answers. In each round, workers see a sentence from a review and are asked to select a single word that expresses sentiment, then to indicate whether this word has a positive or negative polarity. Therefore, this task becomes similar to that of Musat et al. [85], who also asked workers to select sentiment words from sentences. We also very briefly try to extend the game such that,



Figure 5.1: Volunteer participation. Animal puzzles game in the design exploration phase

once a worker successfully solves several rounds, she is invited to graduate to an advanced level that requires her to select more than one word, if necessary. However, we eventually decide to investigate this more complex task in a separate design. Finally, if a sentence does not express an opinion or a worker is unsure how to answer, there is also the option to skip a round.

To engage workers, we rely solely on intrinsic motivation and design our task as a game. When workers first land on our game page, they see a short textual introduction. This invites workers to play an educational game about sentiment words. It explains that, by doing so, they will be helping computers understand opinions and thus they will be contributing to science. If interested, workers can click on a button and land on our main game page (Figure 5.1). As for the context acquisition task, we use a game metaphor that couples a scoring mechanism with animal puzzles. Workers are instructed that their goal is to solve a big mission, which consists of unlocking all the pieces of an animal puzzle that is assigned to them. They are also instructed that, in order to do this, they need to solve smaller missions, which consist of submitting answers and earning points. At the end of a big mission, workers receive more details about the animal portrayed in the unlocked puzzle, then start a new mission with another puzzle. To further enhance the game feel, we try to make the interface more attractive and colorful. For our game art, we used:

- Coins icon by Antialiasfactory¹, under Creative Commons licence².

¹<https://www.iconfinder.com/icons/57737>

²<https://creativecommons.org/licenses/by-nc/3.0>

Chapter 5. Sentiment Knowledge Acquisition with Volunteers

- Medal icon by FatCow Web Hosting³, under the same licence.

To make sure that workers quickly understand what the task is about, we give up trying to explain the rules of the game through a lengthy text tutorial. Instead, as with the game for fine-grained annotation acquisition (Chapter 4), we choose to embed very short instructions directly in the game interface. These instructions teach workers: how to solve the big mission, that is, how to uncover all the pieces of the animal puzzle by earning points; and how to solve the small missions, that is, how to earn points by constructing answers. In addition, during the first round, we complement the text instructions with a video tutorial that demonstrates how the answer in that particular round should be constructed and how to manipulate the game controls. Finally, we make the answer construction controls more intuitive, by allowing workers to select a word by clicking or tapping on it.

Village Explorer

For our second task design, we still require workers to provide simple answers. Similar to the previous design, an answer consists of a single expression (most of the times, an individual word), along with its polarity. This task is also structured in rounds. However, a round no longer displays a full sentence, but only a single expression. A worker is asked to label this phrase as having a positive, neutral, or negative polarity. Therefore, this task design becomes similar to that of Hong et al. [36], who also asked workers to annotate individual words with their polarities.

To engage workers, we again design the task as a game. Contrary to our first task, we give up using an introductory page that invites workers to play and contribute to science. This is because our initial trials hinted that this may in fact discourage some of the workers from participating, given that it is a barrier between them and the actual game. Instead, we insert a similar, but shorter text introduction directly in our main game interface. We use a game metaphor in which the worker is a small explorer that needs to solve missions (Figure 5.2). Each mission consists of travelling through a forest in order to visit several villages. When a village is visited, a worker receives a round to solve. With each answer submitted, she receives some points. Aiming to increase worker motivation, we try to make a mission more challenging by setting a time limit with the help of a counter. Depending on whether a mission is completed within the given time limit and on whether the total points earned exceed a certain threshold, the worker is notified whether she has won or lost the mission. In both cases, she is encouraged to start a new mission by pressing any key. A substantial part of this metaphor is borrowed from and implemented based on two tutorials that introduce Crafty - a game engine for JavaScript⁴: one written by Darren Torpey [114] and one by Louis Stowasser [111]. These tutorials explain how to build a very simple game involving a character that can be moved around such that it avoids obstacles and visits villages, or admires flowers.

³<https://www.iconfinder.com/icons/36193>

⁴<http://craftyjs.com>

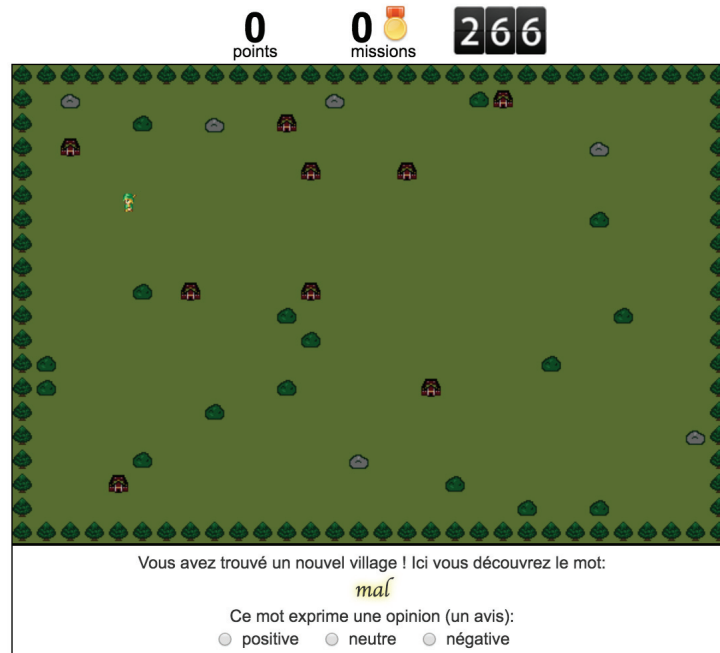


Figure 5.2: Volunteer participation. Village explorer game in the design exploration phase

For our game, we use the same art as in the tutorial of Darren Torpey. This includes:

- Explorer sprite, based on a sprite map created by Antifarea⁵, under Creative Commons licence⁶.
- Forest sprite (tree, bush, stone, village) based on a sprite map created by Sharm⁷, under the same licence.
- jQuery counter plugin by Sophilabs⁸.

To make sure workers understand the task, we rely on very short instructions, which explain that the purpose of the game is to visit villages and discover words. The instructions also explain how to manipulate the explorer character by using the keyboard. Moreover, during each round, we use short sentences that introduce the newly discovered word and invite the worker to choose one of three possible polarities.

Labyrinth Explorer

For our third task design, we require workers to provide answers that have an increased complexity. We come back to the context acquisition game as a source of inspiration and aim

⁵<http://opengameart.org/content/antifareas-rpg-sprite-set-1-enlarged-w-transparent-background>

⁶<https://creativecommons.org/licenses/by/3.0>

⁷<http://opengameart.org/content/16x16-overworld-tiles>

⁸<https://github.com/sophilabs/jquery-counter>

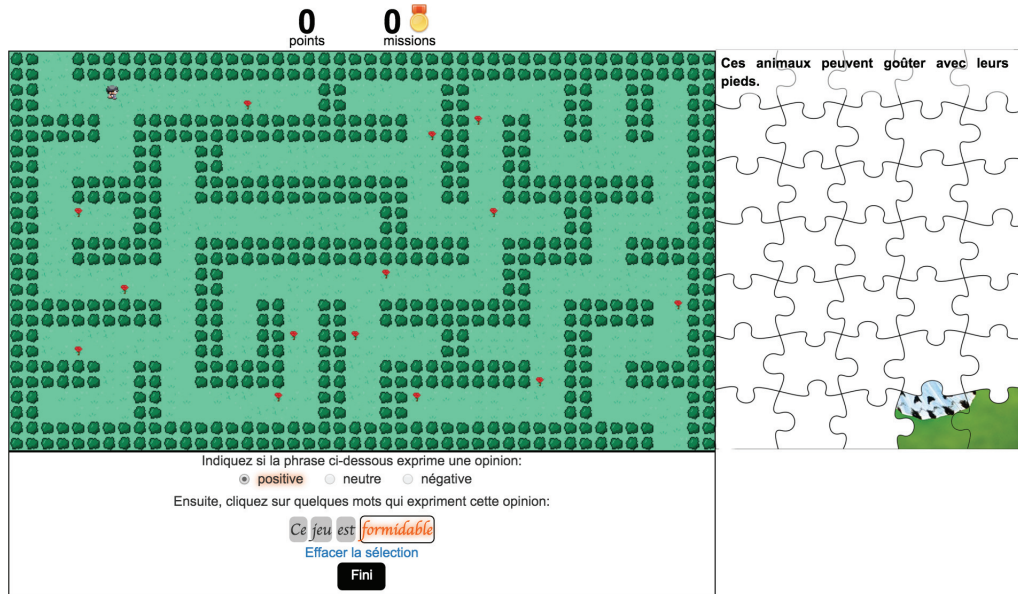


Figure 5.3: Volunteer participation. Labyrinth explorer game in the design exploration phase

to ask workers to provide answers that contain the polarities of words in context. We structure the task in rounds, with each round displaying a sentence selected from an online review. As we cannot afford to rely on a lengthy tutorial that explains what context is and how it can influence the polarities of sentiment words, we do not ask workers to provide answers that explicitly separate sentiment words from their contexts. Instead, we invite workers to first indicate whether the sentence shown to them has a positive, neutral, or negative polarity. In case the worker indicates the sentence is not neutral, we then ask her to click or tap on some of the words that convey this polarity. This is similar to the work of Al-Subaihini et al. [3], who also asked workers to label sentences and to select multiple sentiment words. We thus manage to acquire some contextual knowledge, albeit not structured, when workers decide to select longer work combinations.

To engage workers, we use a game metaphor inspired from the previous two trials. As for the village explorer game, at the top of the main game interface, we insert a short text inviting workers to help science by playing an educational word game. A worker is again a small explorer that solves missions. However, a mission is now more complex, in that the worker needs to collect several flowers whilst navigating through a labyrinth. With each flower collected, the worker receives a round to solve and, with each answer submitted, she receives some points. In search for new ways to increase worker engagement, we remove the timer and instead complement the labyrinth with an animal puzzle displayed on its left side (Figure 5.3). Based on the total points earned at the end of a mission, the worker wins or loses it. In case of a win, the worker receives the solution to the animal puzzle currently assigned to her. Part of this metaphor was also borrowed from and implemented based on the tutorials of Darren Torpey and Louis Stowasser. In addition, we implemented the labyrinth generation based on



Figure 5.4: Volunteer participation. Animal puzzles game in the metaphor comparison phase

the tutorial of Jim Blackler [7]. For our game, we used the same art as in the tutorial of Louis Stowasser. This includes the explorer, grass, flower, and bush sprites.

As for the village explorer game, we ensure workers understand the task using very short instructions. These explain the purpose of the game: collecting flowers in order to discover sentences expressing sentiments. The instructions also explain how to move the explorer character using the keyboard or by clicking or tapping. In addition, during each round, the instructions introduce the newly discovered sentence and guide workers towards constructing and submitting an answer.

Quality Assurance

The instructions we use in these three tasks implicitly influence answer quality, by ensuring workers understand how to solve them. For the animal puzzles and labyrinth explorer tasks,



Figure 5.5: Volunteer participation. Village explorer game in the metaphor comparison phase

which are more complex in that they require workers to select words from sentences, we also try to guide new workers by fixing the first two rounds and by making sure that, for these rounds, workers can only click on the words that need to be selected. The scoring mechanism also ensures quality, by encouraging workers to submit useful answers. During the animal puzzles game, we borrow the same strategy used during the context and annotation acquisition games. That is, we lock workers after 200 rounds submitted or when their average score drops below a certain threshold. However, in all subsequent trials, we give up on this strategy, as we are also interested to see how long workers play for when there are no financial incentives. After the game, we aggregate answers into a sentiment lexicon: by assigning the majority polarity to each word or word combination that has been included by workers in an answer; then by dropping the lexicon elements that have a neutral (ambiguous) polarity.

After we obtain an initial sentiment lexicon, we remove the bad elements - those that harm the sentiment classification performance on a training set with reviews. We achieve this in two steps. We start by removing the features that meet at least one of several criteria: they are infrequent in the training set; they belong to a predefined list of stop words; they occur evenly in both positive and negative reviews; they have a polarity opposite to that recorded in a reference sentiment lexicon (described in more detail in Section 5.3); or they have a polarity that does not reflect the features' frequency distribution in the positive and negative reviews. We then proceed to combine the pruned lexicon with the reference one. We evaluate this

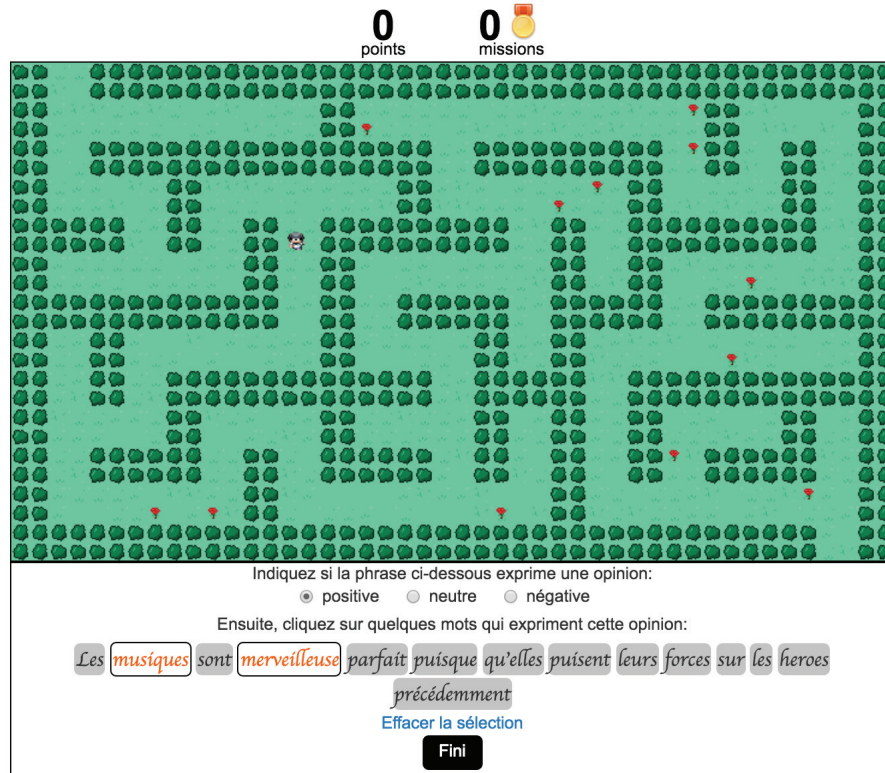


Figure 5.6: Volunteer participation. Labyrinth explorer game in the metaphor comparison phase

extended lexicon on the training set and remove the human-generated features that harm the standalone performance of the reference lexicon. In doing so, we classify reviews by counting the frequency of each positive and negative feature, whether individual words or longer word combinations, and output the polarity label represented by most words.

5.2.2 Game Metaphor Comparison

Once we conclude the exploratory phase, where we aim to figure out what convinces workers to voluntarily take part in our tasks, we also conduct a more structured study, where we aim to compare the effectiveness of the various game metaphors that we propose. To perform this comparison, we need to fix the question that we want workers to solve and vary the game metaphor that goes along with it. Because with our third exploratory trial we discovered that workers can also answer the more complex question of selecting longer sentiment expressions, we fix this as the task we want to gamify. To study what motivates workers to participate, we compare the three game metaphors previously introduced: the animal puzzles, the explorer visiting villages, and the explorer navigating through a labyrinth in search for flowers. Since the task we focus on is more complex, we simplify the village explorer metaphor by removing the timer. This is because we do not want to rush workers into submitting poor quality answers.

Moreover, as we want to separately assess the impact of the animal puzzles and of the labyrinth explorer metaphors, we simplify the third game such that it only displays the labyrinth and no longer shows the puzzle on the left side (Figures 5.4, 5.5, and 5.6 show the final game interfaces). As explained previously, all three games consist of solving missions by submitting answers and earning points. For the animal puzzles metaphor, workers complete a mission when they have managed to unlock all the pieces of the puzzle assigned to them. At the end of a mission, workers receive more details about the animal and can proceed to solve new missions. For the other two games, workers finish the mission when they have managed to visit all the villages and collect all the flowers, respectively. At the end of the mission, workers are notified whether they have won or lost it and are encouraged to start new ones.

Quality Assurance

For all three instances of this task, we take the same three quality measures as during the exploratory phase. Again, the short instructions we embed in the game interface make sure workers understand the task. Moreover, our experience with the second and third exploratory task designs taught us that a video tutorial is not actually mandatory, provided that the instructions and controls are intuitive enough. Therefore, we no longer use such a tutorial in the animal puzzles game. Moreover, all three games start with two training rounds that guide workers towards finding the correct answers. This is achieved by highlighting the correct choices and by disabling all the other options. Note that these training rounds are the same in all three games. What is more, we couple each of these three metaphors with the same scoring mechanism, which assesses whether an answer given by a worker agrees with the answers of previous workers. Finally, after workers have finished the task, we aggregate their answers into a sentiment lexicon, from which we remove the bad elements.

5.2.3 Worker Recruitment

The Google Adwords platform can be used for running ads on Google Search as well as on websites that partner with Google (the Google Display Network). When using this platform, one needs to start by creating a campaign, which consists of one or more ad groups - a set of advertisements along with the set of keywords that trigger their display (impression). Ads are shown on Google Search when users look up these keywords. They are also shown on partner websites whose content is highly relevant for the keywords in an ad group. An important detail is that a campaign can be configured such that ads are shown only to users that come from a set of desired regions (e.g. at the level of countries or provinces). To recruit voluntary workers, we use this platform to advertise our tasks. Because we aim to design tasks that acquire sentiment knowledge in French, we show ads to workers coming from French-speaking territories.

An advertiser is charged per clicks and not per impressions. A campaign can be configured with a daily budget as well as with a maximum amount that one is willing to pay for a click. Each ad receives a quality score, which influences how much the advertiser pays for a click:

the higher the quality of an ad, the less the advertiser needs to spend in order to have her ad shown. This quality score depends on multiple factors, which include the overlap between the keywords and the content of the ad or that of the landing page (the page that the user sees after clicking on the ad). There are other factors as well, like whether the advertised site is optimized for both desktop and mobile (e.g. in terms of loading speed). We keep these issues in mind when choosing our keywords and ad texts and when implementing our tasks.

A more advanced detail is that the platform optimizes ad impressions for clicks: ads are placed such that the number of users who click on them is maximized. However, there is also the notion of conversion: a user who clicks on an ad converts if she engages with the advertised page in the manner expected by the advertiser (e.g. buying the advertised product). These conversions can be tracked and, when they have reached a minimum threshold, a campaign can be configured to maximize the number of users who convert instead of the number of clicks. We use this feature to attract clicks from workers that are likely to engage.

5.3 Empirical Results

We tested our human computation designs using reviews about video games, written in French. We conducted the exploratory and metaphor comparison phases of our study by running online advertisements that linked to our games.

We present our results, structured around several major conclusions that they support.

5.3.1 Dataset

For our experiments, we relied on a corpus with online reviews written in French. The reviews describe video games⁹. A small fraction of these were used during the DEFT competition in 2007 [30], and more were downloaded by us. We split this corpus into three parts. We used one subset to create the rounds in our human computation games. We used a training set to calibrate the sentiment lexicons that we generated from the workers' activity in our tasks. Finally, we used a test set to evaluate the quality of the human-generated sentiment lexicons. We ensured that both the training and the test sets had an equal number of positive and negative reviews. More specifically, these contained: 15,418 and 12,504 reviews, respectively.

5.3.2 Exploratory Phase

In the exploratory phase, we launched the initial versions of the animal puzzles, village explorer, and labyrinth explorer games. We instantiated the rounds in the animal puzzles game with sentences selected from the first subset of video game reviews. For the village explorer game, in its initial version consisting of rounds that displayed individual expressions,

⁹They were downloaded from the website <http://www.jeuxvideo.com>

Chapter 5. Sentiment Knowledge Acquisition with Volunteers

Jeu de lecture et de mots Petit jeu éducatif pour s'entraîner à la lecture lia-jeux.epfl.ch/jeueducatif	Jeu de mots et de lecture Petit jeu éducatif pour s'entraîner au vocabulaire lia-jeux.epfl.ch/jeuemots
---	---

Figure 5.7: Volunteer participation. Examples of advertisements for the games ran in the exploratory phrase

we created a set of phrases based on some of the answers collected with the animal puzzles game. The vast majority of these expressions consisted of individual words. Finally, for the labyrinth explorer game, we again instantiated the rounds with review sentences.

For each of our three games, we created a separate campaign. Each of these campaigns contained a single ad group consisting of several ads and keywords. We advertised educational games for improving reading skills and vocabulary knowledge (Figure 5.7). For example, we created one ad with the title *Jeu de lecture et de mots* (Reading and word game) and description *Petit jeu éducatif pour s'entraîner à la lecture* (Small educational game to practice reading skills). We used matching keywords, like: *jeu éducatif* (educational game), *jeu de lecture* (reading game), or *jeu de mots* (word game). Because our games were targeting French workers, we restricted our campaigns to several French-speaking territories (e.g. France, several regions in Belgium and Switzerland). For each of the three campaigns, we experimented with daily budgets of at most \$60 and with a maximum cost per click of roughly \$0.5. We started each campaign by optimizing ad placement for clicks, and tried to switch to conversion optimization when this option became available. For our purposes, we defined a conversion as a worker solving a round in the game. We mostly used the default option of recording a conversion as a singular event per worker, although we also briefly tried the feature of recording multiple conversions per worker (one for each round she solved in the game). Note that, for the animal puzzles game, which in its initial version included a welcoming page, we also defined an intermediate conversion event, capturing when users clicked on the button for starting the game. However, this turned out not to be an ideal strategy, given that the platform began to show our ads to workers who would abandon the game once they reached the main game interface. In the results that follow, we do not report these intermediate conversion results.

By combining online advertising with human computation games, we obtain an effective recipe for recruiting and engaging volunteers

We ran the three campaigns one at a time, for several weeks each (Table 5.1). We started by running the campaign advertising the animal puzzles game. In total, we spent about \$600 for about 2,900 clicks and a click-through rate of 0.48%. We convinced 214 workers to play, for a conversion rate of 7.4%. These workers provided roughly 3,100 answers, which amounted to an average of 14.4 answers per conversion and a cost of \$0.19 per answer. We subsequently

	animal puzzles	village explorer	labyrinth explorer
period	24.03 - 20.04	26.04 - 01.06	10.06 - 13.07
impressions	601k	521k	758k
clicks	2,881	2,620	3,742
click-through rate	0.48%	0.50%	0.50%
conversions	214	290	306
conversion rate	7.43%	11.07%	8.18%
total cost	\$600	\$631	\$1,186
cost per click	\$0.21	\$0.24	\$0.32
cost per conversion	\$2.80	\$2.18	\$3.88
total answers	3,091	3,607	1,843
answers per conversion	14.4	12.4	6.0
cost per answer	\$0.19	\$0.17	\$0.64

Table 5.1: Volunteer participation. Statistics of the advertisement campaigns we ran in the exploratory phase (in 2015)

ran the village explorer campaign. We spent roughly \$630 dollars for about 2,600 clicks and a click-through rate of 0.5%. We convinced 290 workers to play, leading to a conversion rate of 11.1%. These workers contributed a total of 3,600 answers, giving an average of 12.4 answers per conversion and a cost of \$0.17 per answer. In the third campaign, we ran the labyrinth explorer game. We spent approximately \$1,200 for 3,700 clicks and a click-through rate of 0.5%. We recorded 306 conversions, for a conversion rate of 8.2%. The participating workers submitted a total of 1,850 answers, for an average of six answers and a cost of \$0.64 per answer.

We were able to reach a reasonable number of clicks, and all three campaigns led to similar click-through rates. The latter can be further broken down into search and display rates. The search rates ranged between 1.1 - 1.6% and the display rates between 0.3 - 0.4%. While previous crowdsourcing experiments [43, 59] did not report click-through rates, it is our understanding that rates of under 1-2% are typical for Adwords campaigns [98, 44]. On the other hand, it is difficult to indicate a benchmark for search rates, as these can vary across industries and according to many parameters that one does not have control over [98] (although some sources estimate this rate to be 1.91% across all industries [44]). Nevertheless, the quality score that Adwords attaches to keywords in a campaign can be taken as an indicator for whether their corresponding click-through rates are above or below average. For two of our campaigns (animal puzzles and labyrinth explorer), the keywords *jeu de mots* and *jeu de lecture* reached quality scores of 5/10¹⁰, which can be broken down into three aspects: average click-through rates, above average ad relevance, and below average landing page experience. On the other hand, the keyword *jeu éducatif* had a score of 3/10, given that its click-through rate was assessed as below average, along with the landing page experience. This lower performance, especially compared to the other two keywords, might be due the fact that, at the time of these campaigns, users who searched for educational games expected something more sophisticated

¹⁰As checked on 7th of September 2016

than a word game, and were thus discouraged by the text of our ads. Of course, there are other parameters that could have also contributed, like the average position of the ads when triggered by this keyword. Nevertheless, the phrase *jeu éducatif* brought a substantial number of clicks, which is why we kept it. Overall though, based on all these clues, we believe that we were able to reach reasonable click-through rates.

The three games also led to reasonable conversion numbers and rates. Kobren et al. [59] do not report conversion rates. On the other hand, Ipeirotis and Gabrilovich [43] reached a conversion rate of 35%, increasing from 20% to 50% over a one month period (as a result of continued use of conversion optimization, coupled with feedback on the workers' quality of contribution). On the other hand, other sources [44, 53] aggregated conversion rates for Adwords campaigns across industries and found the average rate to be around 2.3% - 2.7%. Therefore, while our campaigns performed under that of Ipeirotis and Gabrilovich, it seems that, looking at the big picture, our conversion rates were well above average. Moreover, we believe that further optimizing the campaigns is likely to further bridge the gap with respect to [43].

Separately looking at the three conversion rates, the animal puzzles gave the lowest. This might be because this task started with the welcoming page inviting workers to play, and not with the main game page. It could also be that the game metaphor was not appealing enough. The village explorer game led to the highest conversion rate. A possible explanation is that, in this task, we removed the welcoming page and directly showed the game interface. Another explanation could be that this game metaphor was more appealing. Finally, with the labyrinth explorer game, the conversion rate dropped again. This might be because workers thought the game metaphor, combining a labyrinth with animal puzzles, was too complex. What might have also influenced the three conversion rates is the fact that each game assigned workers different tasks, of varying complexities. Furthermore, the games were advertised in different campaigns, each with slightly different parameters. Finally, the three game implementations were optimized for desktop and mobile devices to different extents (see more below), which could have also influenced the conversion rates (e.g. by impacting keyword quality scores and thus the number and position of ad impressions; by having an effect on the workers' experience once they landed on our page). However, even though we cannot make a direct comparison between the three conversion rates, we can nevertheless conclude that these are reasonable statistics.

From the three campaigns, we can also observe that workers played for a reasonable number of rounds. On average, they engaged with the animal puzzles and village explorer to the same extent, by submitting a similar number of answers. For the labyrinth explorer, the average number of answers was substantially lower. This may have to do with the fact that this game had an increased complexity, both in terms of the questions it asked and its metaphor. We should again point out that what could have also contributed to this difference in behavior is that the three advertisement campaigns had different parameters. As a result, compared to Ipeirotis and Gabrilovich [43], who reported that workers submitted 9.2 answers on average,

our first two campaigns were more efficient, whereas the third one was less productive (Kobren et al. [59] did not report these statistics). With respect to our context acquisition game, which had an increased complexity (as it required workers to explicitly identify context features) and for which we recruited paid workers, these statistics were substantially lower (the workers that we recruited on Amazon Mechanical Turk submitted eighty to ninety answers on average). It might be that combining more evolved games with an Adwords campaign that is optimized for conversion during a longer period would attract workers that are willing to play longer. This intuition is supported by the fact that we had thirteen workers that played at least eighty rounds, with some submitting as much as 200 or 288 answers. However, the large gap in worker behavior also hints that financial incentives probably have a say in how long workers interact with the game. Nevertheless, our reasonable conversions rates, coupled with the fact that workers played at least a few rounds, shows that we were able to effectively recruit and engage voluntary workers to play our word games.

In terms of the cost per click, these were similar for the the animal puzzles and village explorer campaigns, but higher for the labyrinth explorer. Compared to the work of Ipeirotis and Gabrilovich [43], who reported an average cost per click of \$0.037, our expenses were five to eight times higher, depending on the campaign (Kobren et al. [59] did not report these costs). However, our costs seem within reasonable bounds with respect to other sources [44], who reported costs of \$2.32 and \$0.58 for the search and display network, respectively, across industries. We expect that further fine-tuning of our campaigns would help to decrease our expenses (e.g. improving the landing page experience would increase the quality score of our keywords, which in turn would lower the cost per click). However, one does not have full control in optimizing this cost, as how much one needs to bid for a keyword also depends on what the competition is bidding.

The combined effect of these three aspects (the incurred costs per click, the recorded conversion rates, and the observed worker engagement) was that the costs per answer were similar for the animal puzzles and village explorer, but roughly three times higher for the labyrinth explorer. On the other hand, Ipeirotis and Gabrilovich [43] reported an average cost per answer of \$0.012. Moreover, in our context acquisition experiments on Amazon Mechanical Turk we paid roughly \$0.02 per answer. Therefore, our Adwords costs were substantially higher than both of these reference points. This means, at the moment, our Adwords costs are not competitive enough and probably too expensive for practical applications. Therefore, one should continue to look for ways in which the cost per answer can be improved, by jointly working on the three fronts mentioned above. Firstly, by lowering the cost per click (for as much as this can be controlled), for example by improving the landing page experience. Secondly, one should continue to increase the conversion rate: on the one hand, by making the games more appealing; on the other hand by further running conversion optimization. Finally, one should also strive to improve engagement, such that workers stay longer in the game and perhaps even come back to it multiple times. Again, on the one hand, this could be achieved by making more evolved games. On the other hand, one could also more seriously try to record each answer as a separate conversion in Adwords, such that ad placement is optimized in order to

	animal puzzles	village explorer	labyrinth explorer
same polarity	343	155	85
opposite polarity	33	12	21
accuracy	91.22%	92.81%	80.18%

Table 5.2: Volunteer participation. Overlap between the three lexicons generated during the exploratory phase and the reference lexicon

maximize the number of answers per conversion. Nevertheless, we believe that these initial results are encouraging, and that it should be possible to further bridge the cost gap.

Volunteers submit good quality knowledge

We also studied the sentiment classification performance of the answers that workers submitted while playing our three games. As a starting point, we used an existing sentiment lexicon in French that we obtained from Dermouche et al. [24]. This lexicon contains 3,362 individual words and several longer word combinations, for a total of 3,927 elements. We separately aggregated the answers that we acquired with the three human computation games, respectively. We thus obtained three human-generated sentiment lexicons. The first lexicon consisted of 1,406 elements. These were mostly individual words and only 23 elements were longer combinations, acquired during our brief attempt to invite workers to play an advanced level in the animal puzzles game. The second lexicon was, with one exception, comprised of individual words and contained 557 items. Finally, the third lexicon contained a more substantial number of longer word combinations. More specifically, it contained 434 individual words and 367 longer phrases. Note that, for the animal puzzles and labyrinth explorer games, the vast majority of these longer word combinations were constructed by workers who explicitly clicked on more tokens. For the rest, we suspect they resulted from workers clicking on a longer phrase that was erroneously displayed as a single token in the interface.

We started by analyzing how the human-generated lexicons overlapped with the reference lexicon. For each of these lexicons, we verified how many of the items they contained were also present in the reference lexicon. In addition, for each element in the overlap, we checked whether workers indicated the same polarity as in the reference lexicon. We learned that, for the animal puzzles lexicon, roughly 380 words were present in the reference lexicon, and that 91% of these had the same polarity (Table 5.2). For the village explorer lexicon, there were approximately 170 words in the intersection, and 93% had the correct polarity. Finally, for the labyrinth explorer lexicon, there were 110 words in the overlap, and for 80% their polarities coincided with those in the reference lexicon. Therefore, in the majority of cases when workers submitted sentiment words that were captured in the reference lexicon, they were able to indicate correct polarities. This is a first indicator that workers managed to submit good quality answers. In addition, only 10% to 30% of the elements in the human-generated lexicons were part of the reference lexicon. This means that workers were also able to discover new sentiment features.

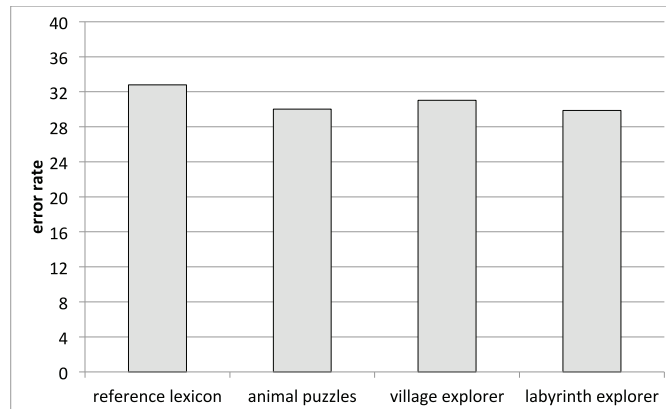


Figure 5.8: Volunteer participation. Sentiment classification performance in the exploratory phase

Jeu éducatif de mots
 Petit jeu éducatif pour s'entraîner
 les connaissances des mots
lia-jeux.epfl.ch/jeueducatifmots

Figure 5.9: Volunteer participation. Advertisement text used in the metaphor comparison phase

We then studied whether the workers' answers could improve the sentiment classification performance of the reference lexicon. We started by removing the bad elements from the three human-generated lexicons. After this step, the three lexicons contained: 1,088, 475, and 485 elements, respectively. We then evaluated the performance of the reference lexicon on the test set. Finally, we combined the reference lexicon with each of the human-generated lexicons and assessed the resulting performance. The reference lexicon gave an error rate of 32.80% (Figure 5.8). When it was combined with the three generated lexicons, its error decreased to 30.01%, 31.03%, and 29.86%, respectively. Therefore, all three lexicons were able to decrease the error of the reference sentiment lexicon. This is further proof that workers were able to contribute with good quality answers that managed to improve the performance of an existing sentiment lexicon in French.

There is a need for adaptation for desktop and mobile

As we were running these three task designs, we learned that workers were accessing our pages from both desktop and mobile devices, as well as from various browsers. We thus realized that our implementations needed to cover as many of these cases as possible. Given that a substantial amount of traffic came from mobile devices, we especially focused on optimizing our tasks for phones and tablets. While the animal puzzles game functioned on mobile without needing major interventions, the village explorer game only worked for desktops. This is because workers did not have the option to move the explorer by clicking or tapping, since

the implementation of this feature was not trivial. During our third trial, we were able to also adapt the labyrinth explorer game, by enabling this option. However, we decided to hide the animal puzzles from the left side of the labyrinth for the mobile traffic, since mobile devices have screens that are typically smaller. In general, we also tested that our implementations functioned correctly in several browsers. Moreover, we learned that the load speed of task pages needed to be optimized by reducing the size of the JavaScript and CSS code, as well as that of the game art.

5.3.3 Game Metaphor Comparison

Once we completed the exploratory phase, we continued by running the more structured comparison between our three human computation tasks. With this comparison, we aimed to more systematically investigate what game metaphor convinces workers to play. We thus removed some of the other factors that could have interfered with the workers' conversion and their engagement with the game thereafter. Therefore, we ran the final versions of our human computation games, which asked workers to solve the more complex question of selecting longer sentiment features. We instantiated the three games with the same set of rounds. As much as possible, we tried to similarly optimize all three task implementations, for both desktop and mobile. Moreover, to eliminate fluctuations caused by running one campaign at a time, we attracted workers by advertising these games in parallel, using the same Adwords campaign. For this purpose, we chose one of the campaigns that we had used during the exploratory phase. The advantage was that this campaign had already accumulated some conversions, which allowed us to advertise our tasks by optimizing for conversions from the start. For the chosen campaign, we modified the existing ad group by creating three new ads. These had identical texts and linked to the three games, respectively (Figure 5.9). We coupled the ads with the keywords *jeu éducatif* and *jeu de mots*. We had ran the campaign for a few weeks when we noticed that the three ads had not been shown evenly, but that the Adwords platform was rotating them in order to maximize the number of clicks (this is the default setting). We thus reset the data in the games and continued to run the campaign, this time by ensuring that the three ads were rotated evenly.

We first looked at the statistics of the two campaign runs. First of all, in terms of click-through rates, our results resembled those from the game metaphor exploration phrase. Overall, for the two campaign runs, we recorded a search rate of 1.21% and a display rate of 0.57%. Moreover, the two keywords got assigned quality scores of 5/10 and 6/10, partly based on click-through rates estimated as average and above average, respectively. Secondly, in terms of conversion rates, these were similar to what we recorded during the exploration phase, meaning that they were substantially lower than what Ipeirotis and Gabrilovich [43] reported, but well above the conversion rate across industries [44, 53]. Finally, in terms of costs per click, these were similar to the expenses we incurred in the exploration phase. The combined effect of these observations was the average cost per answer was still relatively prohibitive.

	animal puzzles	village explorer	labyrinth explorer
period	15.02.2016 - 02.03.2016		
impressions	135k	99k	370k
impression rate	22.37%	16.41%	61.21%
clicks	806	1,051	2,378
conversions	112	116	204
conversion rate	13.90%	11.04%	8.58%
total cost	\$134	\$217	\$453
cost per click	\$0.17	\$0.21	\$0.19
cost per conversion	\$1.20	\$1.87	\$2.22
total answers	712	827	1,153
answers per conversion	6.36	7.13	5.65
cost per answer	\$0.19	\$0.26	\$0.39

Table 5.3: Volunteer participation. Statistics of the first advertisement campaign run in the metaphor comparison phrase

In more detail, the statistics of the first campaign run (Table 5.3) indeed showed that the ads corresponding to the three games were impressed disproportionately: the labyrinth explorer was advertised most, followed by the animal puzzles, and the village explorer. Because ad impressions were not evenly distributed, there were also disproportions in the number of clicks, conversions, and total answers acquired: the labyrinth explorer led, followed by the village explorer, and the animal puzzles. However, even though the latter attracted the fewest clicks, it gave the highest conversion rate: 13.9%. The village explorer game followed, with a conversion rate of 11.0%. However, the difference between the two games was not statistically significant, according to a two-proportion z-test. Finally, the labyrinth explorer game had the lowest conversion rate, of 8.6%. The difference with respect to the other two games was statistically significant. Since the animal puzzles had the highest conversion rate, it led to the smallest cost per conversion and per answer. Finally, converted workers engaged with the three games similarly, by submitting between five to seven answers on average. These variations were not statistically significant according to an unpaired two-tailed t-test.

From the statistics of the second campaign run (Table 5.4), we could see that the three game ads were more evenly shown, which led to a similar number of clicks. However, the animal puzzles game led in terms of conversions and had the highest conversion rate, of 15.2%. The labyrinth and village explorer games followed, with conversion rates of 10.3% and 9.0%. The differences between the top conversion rate and the other two were statistically significant. However, the difference between the conversion rates of the two explorer games was not. Given that the animal puzzles gave the highest conversion rate in this run as well, it again led to the smallest costs per conversion and per answer. In terms of worker engagement, this was similar to what we observed in the first run: the average number of answers per converted worker ranged between five and eight, but the variations between the three games were not statistically significant.

Chapter 5. Sentiment Knowledge Acquisition with Volunteers

	animal puzzles	village explorer	labyrinth explorer
period	03.03.2016 - 18.03.2016		
impressions	153k	143k	165k
impression rate	33.29%	31.18%	35.53%
clicks	992	981	1,068
conversions	151	88	110
conversion rate	15.22%	8.97%	10.30%
total cost	\$216	\$218	\$228
cost per click	\$0.22	\$0.22	\$0.21
cost per conversion	\$1.43	\$2.47	\$2.07
total answers	1,269	501	715
answers per conversion	8.40	5.69	6.5
cost per answer	\$0.17	\$0.44	\$0.32

Table 5.4: Volunteer participation. Statistics of the second advertisement campaign run in the metaphor comparison phrase

Finally, as a side note, we have seen that, when using the default Adwords option of optimizing ad rotation to maximize the number of clicks, the ads corresponding to the three games were shown unevenly, with a strong bias for the labyrinth explorer. Looking in even more detail at the campaign statistics, we noticed that, on the first day, the three ads were shown relatively evenly (with impression rates of 36%, 31.14%, and 32.85%, respectively) and scored 38, 20, and 28 clicks, respectively. However, on the second day, the bias was already present (impression rates of 38%, 12.41%, and 49.57%) and translated to 62, 25, and 106 clicks respectively. Given that, on the second day, due to the impression bias, the labyrinth explorer gathered most of clicks, it is easy to see how this bias got propagated even further, since ad rotation was scheduled to maximize the number of clicks. However, this bias was not present on the first campaign day, when it was actually the animal puzzles who scored the most clicks. Our hypothesis is thus that the Adwords platform introduced the bias from the second day due to the fact that it considered the third game to offer a better landing page experience (the only variable that was different for the three ads). This is not fully obvious to us, given that we tried as much as possible to optimize all three game implementations. In the second campaign run, though, we were able to remove this bias by instructing the Adwords platform to rotate ads evenly. However, this came with a slight increase of the average cost per click.

To summarize, with the default ad rotation option, one loses control over how ads are shown to users. If the ads point to web pages that Adwords does not perceive as similarly optimized in terms of user experience, this is going to lead to a bias of a few ads over the others. In turn, this means that, by receiving less focus, some ads might not be in the position to gather enough statistics to allow us to draw strong conclusions. On the other hand, if all the pages are similarly optimized, we expect that this bias should no longer occur. However, this is not a straightforward feat, and it might require the intervention of an expert in search engine and online advertising optimization. The alternative is to modify the default ad rotation

	animal puzzles	village explorer	labyrinth explorer
first campaign run			
individual words	151	285	261
word combinations	244	163	279
size after qa	241	242	342
second campaign run			
individual words	265	152	207
word combinations	470	85	138
size after qa	492	167	206

Table 5.5: Volunteer participation. Statistics of the lexicons generated during the metaphor comparison phase

	animal puzzles	village explorer	labyrinth explorer
first campaign run			
same polarity	35	34	52
opposite polarity	8	13	10
accuracy	81.40%	72.34%	83.87%
second campaign run			
same polarity	57	34	38
opposite polarity	16	6	12
accuracy	78.08%	85.00%	76.00%

Table 5.6: Volunteer participation. Overlap between the lexicons generated during the metaphor comparison phase and the reference lexicon

setting, which means we regain some control over the experiment. However, if the pages are not similarly optimized, this in turn means, as we have seen, that the cost per click will also increase. Therefore, the implication of regaining some of the control are that either one would have to use more expertise in how to optimize the landing pages, or one would need to accept the increase in expenses that comes with forcing uneven ad rotation.

The animal puzzles metaphor might be the most effective in converting workers

Even though the relative order of the three conversion rates was not identical in the two campaign runs, we can, to some extent, conclude that the animal puzzles were a better suited metaphor for engaging workers: in both campaigns, the animal puzzles outperformed the labyrinth explorer in terms of conversion rate (both results were statistically significant); similarly, it beat the village explorer in both campaigns (however, only the difference in the second campaign was statistically significant). Besides the game metaphors, what could have also interfered with the three conversion rates was the fact that the animal puzzles game directly showed a round to a worker, whereas the explorer games requested the worker to first take some action and visit a village or collect a flower, respectively. However, we can rule out this hypothesis: our statistics show that roughly 22% of the workers that clicked on the

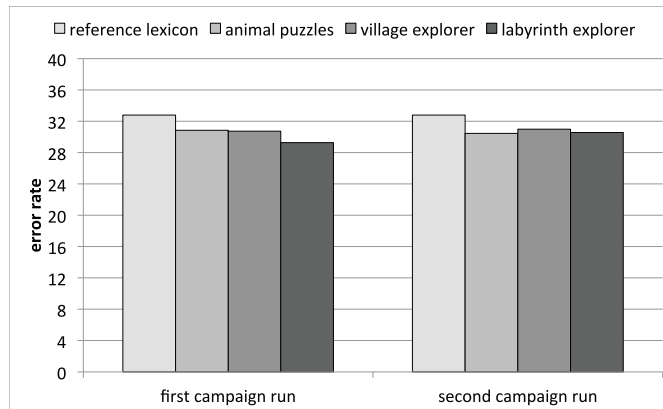


Figure 5.10: Volunteer participation. Sentiment classification performance in the metaphor comparison phase

corresponding ads went as far as this step, but they gave up on solving the first round shown to them. This suggests that these explorer metaphors perhaps mislead some workers into expecting something else from the game, and that is why they abandoned the first round. Moreover, the difference between the village and labyrinth explorer games was less conclusive, so we cannot state with confidence whether one is more attractive than the other. This perhaps also shows that adding a labyrinth to the explorer game metaphor does not complicate it to the extent that the conversion rate suffers significantly. Finally, the results of these two campaigns show that, even when workers are asked to solve more complex questions, we can still reach conversion rates as high as 15%, which is higher than what we obtained during the exploratory phase. This could mean that, provided a suitable game metaphor is chosen, workers can convert even if the question is more difficult. However, what could also account for the higher conversion rate is that we started running the campaign from an already optimized state. To conclude, in terms of worker interaction with the three games, we are to some extent confident that the animal puzzles game is a reasonable choice for convincing workers to engage with our tasks. In addition, the optimal metaphor could still engage workers even when they were asked to submit more complicated answers.

The game metaphor has no significant impact on how long the workers stay in the game

In terms of the average number of answers per worker, there was no statistically significant difference with respect to how workers interacted with the games once converted. One possible reason could be that the game metaphors we proposed are sufficient to convince workers to play several rounds, but not enough to keep workers in the game for a longer time, such that we do get to notice significant differences. This is to some extent understandable for the explorer games, where there is no variation from one mission to another (apart from the randomly placed villages and flowers, and the randomly generated labyrinth). However, this is less intuitive with respect to the animal puzzles game, given that, once workers solve a puzzle, they get a new one. It is possible though that this was not intuitive enough for the workers

playing the game, which is why some quit after unlocking sufficient pieces of the first puzzle assigned to them. This first theory is consistent with the remark that more evolved game metaphors could actually impact the play duration. Another option is that the complexity of the question asked dominates the enjoyment that stems from the game elements (especially if children are playing our games), which is why people get tired and drop out at about the same rate in all three games. One clue sustaining this is the fact that, for the animal puzzles and the village explorer, the average number of answers per conversion were lower than those recorded during the exploratory phase, when the questions asked were simpler (these differences were statistically significant according to an unpaired two-tailed t-test). If that is the case, one could improve the task by making it more of a gradual experience, starting from simple questions and slowly advancing to more complex ones. Finally, another possibility is of course that the differences in the number of answers submitted could become statistically significant if more workers played the game and we gathered more data.

The choice of game metaphor does not influence the quality of the workers' answers

We analyzed the sentiment classification performance of the answers we collected during the two campaigns runs. We aggregated the results obtained during each run and for each of the three games, respectively (Table 5.5). We then studied the overlap between the reference lexicon and the generated ones (Table 5.6). As for the exploratory phase, we learned that, for both campaign runs, workers had a high accuracy when indicating the polarities of words that belonged to the reference lexicon. However, we could not find any statistically significant difference between the workers' accuracy in the three games, for neither of the two campaign runs. This hints that the quality of the workers' answers does not depend on the choice of the game metaphor, and that the latter only impacts which workers convert. Finally, we again noticed that workers were able to select many features beyond the reference lexicon.

We proceeded to remove the bad answers from the human-generated lexicons (Table 5.5 shows their sizes after pruning). We then analyzed whether these lexicons can improve the existing one. In doing so, we recorded results similar to those in the exploratory phase (Figure 5.10): all the generated lexicons helped decrease the error rate of the reference one. These differences were statistically significant according to a paired two-tailed t-test. We also studied whether there was any statistically significant difference in the performance of the lexicons obtained in each campaign run. For the first campaign, we found that, when it extended the reference lexicon, the knowledge acquired with the labyrinth explorer game was significantly better than the other two. However, we were not able to reproduce this result with the second campaign run, when no knowledge set was significantly better than the other. We thus reinforced the intuition that the choice of game metaphor does not have a noticeable impact on the quality of the resulting sentiment lexicon.

5.4 Conclusions

So far, we have studied how human computation can be used to collect commonsense knowledge that improves performance on sentiment analysis problems. We motivated workers to participate through both enjoyment and payment, which proved to be a successful recipe for recruiting and engaging workers that were willing to contribute with good quality knowledge. We ran our tasks with texts in English, and recruited workers on a paid crowdsourcing platform, whose demographics ensured that these were familiar with the language. In this chapter, we aimed to investigate whether we could find a task setup that worked as effectively, but did not rely on financial incentives. If done properly, this would allow to collect sentiment knowledge for languages that are less familiar to the pool of workers on paid crowdsourcing platforms.

We showed that, by combining online advertising with games, we obtain an effective solution for recruiting and engaging voluntary workers that contribute with good quality knowledge for sentiment classification

We first conducted an exploratory phase where we designed three tasks: each wrapping a slightly different question around a different game metaphor. We populated these tasks with texts written in French and recruited voluntary workers by running online advertisements that were placed in order to maximize conversion. Each campaign attracted a reasonable number of clicks. Moreover, a reasonable fraction of the workers clicking on our ads were convinced to participate. Finally, the converted workers provided answers that improved the sentiment classification performance of an existing sentiment lexicon in French.

We hinted what game metaphor might lead to the optimal conversion rate. We further proposed that the choice of game metaphor does not noticeably influence the time spent in the game nor the quality of the contribution, once a worker converts

We also ran a second phase where we systematically compared the effectiveness of three game metaphors in convincing workers to convert. We compared an animal puzzles metaphor with two metaphors instructing workers that they are explorers in search of villages or flowers. In the former, workers unlocked puzzles by solving questions. In the latter, workers received questions as they reached villages or collected flowers. We ran these three games in parallel, in the same advertising campaign. With reasonable confidence, we learned that the animal puzzles metaphor led to the highest conversion rate. However, we also learned that, regardless of the game metaphor used, once a worker converts, the game elements do not have a significant impact in the size of her contribution. Similarly, the game metaphor does not impact the quality of the knowledge acquired: all three games gave knowledge that improved the existing lexicon to a similar extent.

We have thus shown that human computation can systematically improve sentiment analysis, even with voluntary workers.

6 Conclusions

6.1 Summary

Many tasks in artificial intelligence require commonsense knowledge. We illustrated this issue on the sentiment analysis problem, for which a good performance can be achieved on texts that are relatively limited in scope, but not on broad corpora with texts from multiple domains. We studied two sub-problems: document-level sentiment classification and fine-grained opinion extraction. We identified that sentiment classification requires knowledge about the contexts impacting the polarities of sentiment words: this would enable a single classifier to handle a broad domain in a way that reproduces the performance of multiple classifiers specialized on narrow parts of that domain. We also identified that opinion extraction requires multiple fine-grained annotations for texts on varied topics: this would enable an extraction model to also perform well on domains it is unfamiliar with. We explained that context is hard to learn from data, but that humans can easily spot it in texts, using their common sense. We also hinted that, while fine-grained annotations have been tedious to obtain with traditional approaches, it is not necessary to train annotators with detailed manuals and paper exercises. On the contrary, humans can spot the relevant passages of text based on their common sense. We thus sought to use human computation to acquire knowledge that helps sentiment analysis scale to broad domains and generalize further beyond that.

We discussed the main concerns in designing tasks that can effectively collect sentiment knowledge. We proposed to recruit workers on paid crowdsourcing platforms and to engage them by combining payments with entertainment, in games played for money. We aimed to gather answers in a focussed way, while still allowing workers to make complex decisions. We thus designed our games in rounds, in which workers saw review sentences and had to highlight the relevant passages of text: either sentiment words along with meaningful contexts, or opinion expressions and their corresponding targets. Another concern was making sure participants understood the task, and we achieved this with interactive tutorials that tested them with quizzes. Finally, we aimed to effectively control quality, which we achieved through intelligent scoring mechanisms that rewarded useful answers: agreeing with the common

judgement of many workers; and having potential to improve performance.

We employed our games to acquire knowledge from reviews written in English, for multiple product and service categories. We showed that human-generated context helped lexicon and supervised classifiers scale to a broad domain. We also showed that human-generated annotations, coupled with a supervised extraction model that better incorporated syntactic features, helped to improve performance on unfamiliar domains. We concluded that combining games with paid crowdsourcing platforms is an effective recipe for acquiring sentiment knowledge.

We also inquired if tasks could be effectively designed such that workers are recruited outside the crowdsourcing platform and motivated without financial rewards. We proposed that this can be achieved by advertising tasks online and by designing them with only enjoyment in mind. An advantage of this setup is that it can target workers of more varied demographics, which allows to collect knowledge for many other languages. This would be more difficult to achieve on paid crowdsourcing platforms, where the universal language is English. To illustrate these benefits, we created games that acquired sentiment knowledge for reviews written in French, and showed that these helped to improve sentiment classification performance¹.

We have thus shown that human computation can deliver a strong performance for sentiment analysis problems.

6.2 Limitations

6.2.1 In Context Acquisition and Integration

As we have summarized above, we have motivated workers by relying on both enjoyment and payment. In our experiments' interpretation, we have provided an intuition that there was no adverse interaction between extrinsic and intrinsic motivation, and that the latter effectively complemented the former (given that we launched our game on a paid crowdsourcing platform, where payments for non-game tasks are the default, and enjoyable tasks are a pleasant bonus). However, beyond this intuition and a post-game survey that elicited the workers' opinion on the task, we did not attempt to more formally quantify the interplay between payment and enjoyment.

In addition, as we have pointed out in our discussions, we noticed that there were some variations in how workers selected context, both in terms of word boundaries and in terms of whether longer features were split into sentiment word and context pairs, or they were submitted as a whole sentiment expression. However, when aggregating these answers, we chose to simply group them by unique phrase and context components. We did not attempt to normalize the overlapping expressions, which means there was some redundancy in the context models we generated. Finally, in studying how context impacts supervised methods,

¹Note that the context-dependent lexicons as well as the sentiment lexicons in French can be downloaded from <http://liawww.epfl.ch/~boia/lexicons.zip>. However, the fine-grained annotations are not publicly available.

we only thoroughly investigated one machine learning approach, a Support Vector Machine. It would, however, be relevant to see how these features would impact other machine learning algorithms that have been recently employed in sentiment analysis.

6.2.2 In Fine-grained Annotation Acquisition and Integration

We have performed our annotation acquisition study on review sentences with bounded complexity. We achieved this by selecting sentences that complied with a predefined word limit and that contained some syntactic patterns known to be indicative of opinions. On the one hand, these restrictions helped us avoid workers getting too confused, not knowing how to annotate opinions and targets in sentences with very complex syntactic patterns. On the other hand, this also meant that the corpus we constructed was not fully representative of how people express themselves in reviews.

Moreover, we did not invite workers to explicitly indicate whether a sentence did not contain any opinions (as we asked them to just skip those texts). Because the training corpus did not contain non-opinionated sentences, the opinion extraction model we proposed was not equipped to handle such cases. In addition, we did not instruct workers to exhaustively annotate all the opinion and target pairs that appeared in sentences. We only required workers to find one such pair. While sentences with multiple pairs are likely to be covered by aggregating answers from several workers, it is still possible that, for some sentences, we only acquired partial annotations. This of course could have hindered the efficiency of our opinion extraction model, which was likely trained on incomplete labels.

Finally, the extraction model we proposed only handled direct opinion and target dependencies. In some cases, we did complement the model's prediction with negations, adjectival or adverbial modifiers, and direct objects. However, it is likely that, even with these additional syntax heuristics, some longer chain opinion and target dependencies were missed.

6.2.3 In Sentiment Knowledge Acquisition with Volunteers

As we have summarized above, we have recruited and engaged volunteers by advertising our games online. Using the Google Adwords online advertising platform was not straightforward, but a learning process, in which we did a lot of experimentation while trying to figure out what works. We tried several keywords, ad texts, and we did multiple iterations with our task implementations. It thus took a while until we found a suitable experimental setup and our campaigns gained momentum. Therefore, if crowdsourcing results are needed immediately, this solution might not be ideal. We showed that we can use online advertisements to dependably recruit volunteers that play for a reasonable number of rounds. However, our incurred costs per answer were not on par with other Adwords crowdsourcing research, nor with our previous paid crowdsourcing campaigns. This was a combined effect of conversion rates and task engagement capacity, both of which were very encouraging but could have been further

improved, in order to help bridge this cost effectiveness gap.

In addition, given the relatively small scale of our exploratory experiments, the size of the sentiment knowledge we acquired from volunteers was several orders of magnitude smaller than the context models we acquired from paid workers. This meant that, while we were able to show that volunteers do provide good quality knowledge, we were not able to reproduce the same dramatic improvements that we achieved with the knowledge acquired from paid workers. Finally, because relying on volunteers implied that we had to give up using detailed tutorials, we decided to avoid explicitly teaching workers what context is. We did acquire knowledge containing the polarities of longer expressions, but these were not explicitly separated in pairs of sentiment words and their contexts, which would be a more useful context structure.

6.3 Future Work

In terms of future work, one should start by addressing the limitations we enumerated above. Regarding our human computation tasks, the paid ones could be more thoroughly studied to formally establish how payment and enjoyment interact. The unpaid tasks could be further optimized in terms of cost effectiveness, by improving conversion rates and worker engagement. In addition, some of our tasks could be extended to elicit more complex answers. On the one hand, the task eliciting fine-grained annotations could be run with more complex sentences, and could elicit non-opinion as well as exhaustive opinion annotations from workers. On the other hand, the tasks relying on volunteers could be adapted to elicit features complying with the more elaborate context structure that we employed in our paid crowdsourcing campaign. In addition, answer aggregation could be improved such that we can better cope with variations across workers (such as variations in word boundaries). Moreover, we could study how human-generated knowledge impacts other machine learning algorithms as well. In particular, for the fine-grained annotation acquisition task, we could investigate how to design models that can also handle longer-chain opinion and target dependencies.

Beyond the limitations that would have to be addressed, one can also point out other avenues for improvement. For example, one could try to extend our methods to less structured texts, such as blog posts, news articles, or editorials. As these texts are likely to have more complex syntax and semantics, it would be interesting to see whether workers can as effectively identify contexts for sentiment words and targets for opinion expressions. Such documents are also likely to express opinions with respect to more than one entity, which makes them more challenging to analyze. For instance, for the sentiment classification problem, this implies that one would have to infer a polarity label for each of these entities. Therefore, to work with such texts, one would need to incorporate a mechanism for identifying the entities discussed and for discriminating which passages refer to which entities. At that point, contextual knowledge could be used to aggregate the sentiments conveyed with respect to each entity.

Secondly, in terms of human computation task efficiency, one could investigate whether this

can be boosted, for instance using active learning techniques. In our tasks, we randomly sampled the rounds shown to workers, but it would be more efficient to proactively show sentences that are likely to contain useful knowledge. This could be an ambiguous sentiment word along with a disambiguating context, a word combination on whose polarity workers disagree, or an opinion on whose target workers cannot reach a consensus. In addition to selecting relevant sentences based on active learning techniques, one could incorporate other feedback loops as well. For example commonsense knowledge could be acquired in iterations. The knowledge from each iteration could be incorporated into sentiment analysis models, whose performance could then be assessed on test sets. This would allow us to identify documents that sentiment models have problems dealing with. Such tricky documents could then be used to generate new rounds in our human computation tasks, allowing us to acquire knowledge that could help rectify these mistakes. Improving round assignment such that knowledge acquisition is sped up would make our tasks more efficient in terms of the effort required from workers. For our paid tasks, this would also make them cheaper. For our unpaid tasks, this should make smaller worker contributions more likely to bring improvements.

Finally, one could investigate whether our techniques can be applied to other problems in sentiment analysis, natural language processing, and artificial intelligence in general.

Bibliography

- [1] ADREEVSKAIA, A., AND BERGLER, S. Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics* (2006), pp. 209–216.
- [2] AGRAWAL, R., AND SRIKANT, R. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases* (1994), pp. 487–499.
- [3] AL-SUBAIHIN, A., AL-KHALIFA, H., AND AL-SALMAN, A. A proposed sentiment analysis tool for modern Arabic using human-based computing. In *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services* (2011), pp. 543–546.
- [4] AL-SUBAIHIN, A. S., AND AL-KHALIFA, H. S. A system for sentiment analysis of colloquial Arabic using human computation. *The Scientific World Journal* 2014 (2014).
- [5] APPLE. Siri. <http://www.apple.com/ios/siri>, 2011. Accessed: 2015-08-31.
- [6] ARAGON, K. How to be a crowdsourcing rock star (the easy way to go viral). <http://blog.crazyegg.com/2013/09/11/crowdsourcing-rock-star>, 2013. Accessed: 2016-03-10.
- [7] BLACKLER, J. HTML5 maze generator demo. <http://jimblackler.net/blog/?p=316>. Accessed: 2016-06-15.
- [8] BLAIR-GOLDENSOHN, S., NEYLON, T., HANNAN, K., REIS, G. A., MCDONALD, R., AND REYNAR, J. Building a sentiment summarizer for local service reviews. In *Proceedings of WWW-2008 workshop on NLP in the Information Explosion Era* (2008).
- [9] BOIA, M., MUSAT, C. C., AND FALTINGS, B. Acquiring commonsense knowledge for sentiment analysis through human computation. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web* (2014), pp. 225–226.
- [10] BOIA, M., MUSAT, C. C., AND FALTINGS, B. Acquiring commonsense knowledge for sentiment analysis through human computation. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence* (2014), pp. 901–907.

Bibliography

- [11] BOIA, M., MUSAT, C. C., AND FALTINGS, B. Constructing context-aware sentiment lexicons with an asynchronous game with a purpose. In *Proceedings of the 15th International Conference on Computational Linguistics and Intelligent Text Processing* (2014), pp. 32–44.
- [12] BOLLEN, J., MAO, H., AND ZENG, X. Twitter mood predicts the stock market. *Journal of Computational Science* 2, 1 (2011), 1–8.
- [13] BREW, A., GREENE, D., AND CUNNINGHAM, P. Using crowdsourcing and active learning to track sentiment in online media. In *Proceedings of the 19th European Conference on Artificial Intelligence* (2010), pp. 145–150.
- [14] BROSS, J., AND EHRIG, H. Automatic construction of domain and aspect specific sentiment lexicons for customer review mining. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management* (2013), pp. 1077–1086.
- [15] CAI, L., AND HOFMANN, T. Hierarchical document categorization with Support Vector Machines. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management* (2004), pp. 78–87.
- [16] CAMBRIA, E., RAJAGOPAL, D., KWOK, K., AND SEPULVEDA, J. GECKA: Game engine for commonsense knowledge acquisition. In *Proceedings of the Florida Artificial Intelligence Research Society Conference* (2015).
- [17] CAMBRIA, E., XIA, Y., AND HUSSAIN, A. Affective common sense knowledge acquisition for sentiment analysis. In *Proceedings of the 8th International Conference on Language Resources and Evaluation* (2012), pp. 3580–3585.
- [18] CHEN, J., CYPHER, A., DREWS, C., AND NICHOLS, J. CrowdE: Filtering tweets for direct customer engagements. In *Proceedings of the International AAAI Conference on Web and Social Media* (2013).
- [19] CHOI, Y., BRECK, E., AND CARDIE, C. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2006), pp. 431–439.
- [20] CHOI, Y., AND CARDIE, C. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2008), pp. 793–801.
- [21] CHOI, Y., AND CARDIE, C. Hierarchical sequential learning for extracting opinions and their attributes. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (2010), pp. 269–274.
- [22] CHURCH, K. W., AND HANKS, P. Word association norms, mutual information, and lexicography. *Computational Linguistics* 16, 1 (1990), 22–29.

-
- [23] CORTES, C., AND VAPNIK, V. Support-Vector Networks. *Machine Learning* 20, 3 (1995), 273–297.
- [24] DERMOUCHE, M., KOUAS, L., VELCIN, J., AND LOUDCHER, S. A joint model for topic-sentiment modeling from text. In *Proceedings of The 30th ACM/SIGAPP Symposium On Applied Computing* (2015), pp. 819–824.
- [25] DING, X., LIU, B., AND YU, P. S. A holistic lexicon-based approach to opinion mining. In *Proceedings of the International Conference on Web Search and Data Mining* (2008), pp. 231–240.
- [26] FAHRNI, A., AND KLENNER, M. Old wine or warm beer: Target-specific sentiment analysis of adjectives. In *Proceedings of the Symposium on Affective Language in Human and Machine* (2008), pp. 60–63.
- [27] GANU, G., ELHADAD, N., AND MARIAN, A. Beyond the stars: Improving rating predictions using review text content. In *Proceedings of the 12th International Workshop on the Web and Databases* (2009).
- [28] GHORBEL, H., AND JACOT, D. Sentiment analysis of French movie reviews. In *Advances in Distributed Agent-based Retrieval Tools*. Springer Berlin Heidelberg, 2011, pp. 97–108.
- [29] GO, A., BHAYANI, R., AND HUANG, L. Twitter sentiment classification using distant supervision, 2009.
- [30] GROUIN, C., BERTHELIN, J.-B., EL AYARI, S., HEITZ, T., HURAUULT-PLANTET, M., JARDINO, M., KHALIS, Z., AND LASTES, M. Présentation de DEFT’07 (Défi Fouille de Textes). *Actes du 3ème Défi Fouille de Textes* (2007).
- [31] GUERINI, M., GATTI, L., AND TURCHI, M. Sentiment analysis: How to derive prior polarities from SentiWordNet. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2013), pp. 1259–1269.
- [32] GUPTA, A. K. Beta distribution. In *International Encyclopedia of Statistical Science*. Springer Berlin Heidelberg, 2011, pp. 144–145.
- [33] HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. The WEKA data mining software: An update. *SIGKDD Explorations* 11, 1 (2009), 10–18.
- [34] HATZIVASSILOGLOU, V., AND MCKEOWN, K. R. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* (1997), pp. 174–181.
- [35] HO, C.-J., CHANG, T.-H., LEE, J.-C., HSU, J. Y.-J., AND CHEN, K.-T. KissKissBan: A competitive human computation game for image annotation. In *Proceedings of the ACM SIGKDD Workshop on Human Computation* (2009), pp. 11–14.

Bibliography

- [36] HONG, Y., KWAK, H., BAEK, Y., AND MOON, S. Tower of Babel: A crowdsourcing game building sentiment lexicons for resource-scarce languages. In *Proceedings of the Companion Publication of the 22nd International Conference on World Wide Web* (2013), pp. 549–556.
- [37] HOWE, J. Crowdsourcing: A definition. http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html, 2006. Accessed: 2016-03-09.
- [38] HOWE, J. The rise of crowdsourcing. <http://www.wired.com/2006/06/crowds>, 2006. Accessed: 2016-03-09.
- [39] HSUEH, P.-Y., MELVILLE, P., AND SINDHWANI, V. Data quality from crowdsourcing: A study of annotation selection criteria. In *Proceedings of the NAACL HLT Workshop on Active Learning for Natural Language Processing* (2009), pp. 27–35.
- [40] HU, M., AND LIU, B. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2004), pp. 168–177.
- [41] IIDA, R., INUI, K., TAKAMURA, H., AND MATSUMOTO, Y. Incorporating contextual cues in trainable models for coreference resolution. In *Proceedings of the EACL Workshop on The Computational Treatment of Anaphora* (2003), pp. 23–30.
- [42] IPEIROTIS, P. Demographics of Mechanical Turk: Now live! (April 2015 edition). <http://www.behind-the-enemy-lines.com/2015/04/demographics-of-mechanical-turk-now.html>, 2015. Accessed: 2016-05-27.
- [43] IPEIROTIS, P. G., AND GABRILOVICH, E. Quizz: Targeted crowdsourcing with a billion (potential) users. In *Proceedings of the 23rd International Conference on World Wide Web* (2014), pp. 143–154.
- [44] IRVINE, M. Google AdWords benchmarks for your industry [new data]. <http://www.wordstream.com/blog/ws/2016/02/29/google-adwords-industry-benchmarks>. Accessed: 2016-09-07.
- [45] JAKOB, N., AND GUREVYCH, I. Extracting opinion targets in a single- and cross-domain setting with Conditional Random Fields. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2010), pp. 1035–1045.
- [46] JIANG, J. A literature survey on domain adaptation of statistical classifiers, 2008.
- [47] JIN, W., AND HO, H. H. A novel lexicalized HMM-based learning framework for web opinion mining. In *Proceedings of the 26th Annual International Conference on Machine Learning* (2009), pp. 465–472.
- [48] JINDAL, N., AND LIU, B. Opinion spam and analysis. In *Proceedings of the International Conference on Web Search and Data Mining* (2008), pp. 219–230.

-
- [49] JOACHIMS, T. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2002), pp. 133–142.
- [50] KENNEDY, A., AND INKPEN, D. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence* 22, 2 (2006), 110–125.
- [51] KESSLER, J., AND NICOLOV, N. Targeting sentiment expressions through supervised ranking of linguistic configurations. In *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media* (2009), pp. 90–57.
- [52] KESSLER, J. S., ECKERT, M., CLARK, L., AND NICOLOV, N. The ICWSM 2010 JDPA sentiment corpus for the automotive domain. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media Data Workshop Challenge* (2010).
- [53] KIM, L. Everything you know about conversion rate optimization is wrong. <http://www.wordstream.com/blog/ws/2014/03/17/what-is-a-good-conversion-rate>. Accessed: 2016-09-07.
- [54] KIM, S.-M., AND HOVY, E. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics* (2004).
- [55] KLEIN, D., AND MANNING, C. D. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics* (2003), pp. 423–430.
- [56] KOBAYASHI, N., IIDA, R., INUI, K., AND MATSUMOTO, Y. Opinion mining on the web by extracting subject-aspect-evaluation relations. In *Proceedings of the AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs* (2006), pp. 86–91.
- [57] KOBAYASHI, N., INUI, K., AND MATSUMOTO, Y. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (2007), pp. 1065–1074.
- [58] KOBAYASHI, N., INUI, K., MATSUMOTO, Y., TATEISHI, K., AND FUKUSHIMA, T. Collecting evaluative expressions for opinion extraction. In *Proceedings of the 1st International Joint Conference on Natural Language Processing* (2005), pp. 596–605.
- [59] KOBREN, A., LOGAN, T., SAMPANGI, S., AND MCCALLUM, A. Domain specific knowledge base construction via crowdsourcing. *NIPS'14 Workshop on Automated Knowledge Base Construction* (2014).
- [60] KU, L.-W., HUANG, T.-H., AND CHEN, H.-H. Predicting opinion dependency relations for opinion analysis. In *Proceedings of 5th International Joint Conference on Natural Language Processing* (2011), pp. 345–353.

Bibliography

- [61] LAFFERTY, J. D., MCCALLUM, A., AND PEREIRA, F. C. N. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning* (2001), pp. 282–289.
- [62] LAFOURCADE, M. Making people play for lexical acquisition with the JeuxDeMots prototype. In *Proceedings of the 7th International Symposium on Natural Language Processing* (2007).
- [63] LAFOURCADE, M., LE BRUN, N., AND JOUBERT, A. Collecting and evaluating lexical polarity with a game with a purpose. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing* (2015), pp. 329–337.
- [64] LAW, E., AND VON AHN, L. Input-agreement: A new mechanism for collecting data using human computation games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009), pp. 1197–1206.
- [65] LAW, E. L., VON AHN, L., DANNENBERG, R. B., AND CRAWFORD, M. TagATune: A game for music and sound annotation. In *Proceedings of the International Conference on Music Information Retrieval* (2007), pp. 361–364.
- [66] LI, F., HAN, C., HUANG, M., ZHU, X., XIA, Y.-J., ZHANG, S., AND YU, H. Structure-aware review mining and summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics* (2010), pp. 653–661.
- [67] LI, S., LEE, S. Y. M., CHEN, Y., HUANG, C.-R., AND ZHOU, G. Sentiment classification and polarity shifting. In *Proceedings of the 23rd International Conference on Computational Linguistics* (2010), pp. 635–643.
- [68] LI, S.-S., HUANG, C.-R., AND ZONG, C.-Q. Multi-domain sentiment classification with classifier combination. *Journal of Computer Science and Technology* 26, 1 (2011), 25–33.
- [69] LIU, B. *Sentiment Analysis and Opinion Mining*. Morgan and Claypool Publishers, 2012.
- [70] LIU, K., XU, L., AND ZHAO, J. Opinion target extraction using word-based translation model. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (2012), pp. 1346–1356.
- [71] LIU, K., XU, L., AND ZHAO, J. Syntactic patterns versus word alignment: Extracting opinion targets from online reviews. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (2013), pp. 1754–1763.
- [72] LU, Y., CASTELLANOS, M., DAYAL, U., AND ZHAI, C. Automatic construction of a context-aware sentiment lexicon: An optimization approach. In *Proceedings of the 20th International Conference on World Wide Web* (2011), pp. 347–356.
- [73] MAKKI, R., BROOKS, S., AND MILIOS, E. E. Context-specific sentiment lexicon expansion via minimal user interaction. In *Proceedings of the International Conference on Information Visualization Theory and Applications* (2014), pp. 178–186.

-
- [74] MANEVITZ, L. M., AND YOUSEF, M. One-class SVMs for document classification. *Journal of Machine Learning Research* 2 (2002), 139–154.
- [75] MANNING, C. D., AND DE MARNEFFE, M.-C. Stanford typed dependencies manual. <http://nlp.stanford.edu/software/lex-parser.shtml>, 2008. Accessed: 2016-06-15.
- [76] MANNING, C. D., SURDEANU, M., BAUER, J., FINKEL, J. R., BETHARD, S., AND MCCLOSKEY, D. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (2014), pp. 55–60.
- [77] MARKOFF, J. In a video game, tackling the complexities of protein folding. <http://www.nytimes.com/2010/08/05/science/05protein.html>, 2010. Accessed: 2016-06-17.
- [78] MAUREL, S., CURTONI, P., AND DINI, L. L’analyse des sentiments dans les forums. *Atelier Fouille des Données d’Opinions* (2008), 51–59.
- [79] MCCALLUM, A. K. MALLET: A machine learning for language toolkit. <http://mallet.cs.umass.edu>, 2002.
- [80] MCCULLAGH, P., AND NELDER, J. A. *Generalized Linear Models*. Chapman and Hall, CRC Press, 1989.
- [81] MELLEBEEK, B., BENAVENT, F., GRIVOLLA, J., CODINA, J., COSTA-JUSSÀ, M. R., AND BANCHS, R. Opinion mining of Spanish customer comments with non-expert annotations on Mechanical Turk. In *Proceedings of the NAACL HLT Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk* (2010), pp. 114–121.
- [82] MILLER, G. A. WordNet: A lexical database for English. *Communications of the Association for Computing Machinery* 38, 11 (1995), 39–41.
- [83] MOHAMMAD, S. M., AND TURNEY, P. D. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence* 29, 3 (2013), 436–465.
- [84] MORSCHHEUSER, B., HAMARI, J., AND KOIVISTO, J. Gamification in crowdsourcing: A review. In *Proceedings of the 49th Hawaii International Conference on System Sciences* (2016), pp. 4375–4384.
- [85] MUSAT, C. C., GHASEMI, A., AND FALTINGS, B. Sentiment analysis using a novel human computation game. In *Proceedings of the 3rd Workshop on the People’s Web Meets Natural Language Processing* (2012), pp. 1–9.
- [86] MUSAT, C. C., LIANG, Y., AND FALTINGS, B. Recommendation using textual opinions. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence* (2013), pp. 2684–2690.

Bibliography

- [87] O'CONNOR, B., BALASUBRAMANYAN, R., ROUTLEDGE, B. R., AND SMITH, N. A. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media* (2010), pp. 122–129.
- [88] OF ABERDEEN COMPUTING SCIENCE, U. CS4025: Syntax, grammar and parts of speech. <http://homepages.abdn.ac.uk/advaith/pages/teaching/NLP/lectures/lec05.pdf>. Accessed: 2016-06-15.
- [89] PAN, S. J., AND YANG, Q. A survey on transfer learning. *IEEE Transaction on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.
- [90] PANG, B., LEE, L., AND VAITHYANATHAN, S. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2002), pp. 79–86.
- [91] POLANYI, L., AND ZAENEN, A. *Computing Attitude and Affect in Text: Theory and Applications*. Springer Netherlands, 2006, ch. Contextual Valence Shifters, pp. 1–10.
- [92] PONTIKI, M., GALANIS, D., PAVLOPOULOS, J., PAPAGEORGIOU, H., ANDROUTSOPOULOS, I., AND MANANDHAR, S. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation* (2014), pp. 27–35.
- [93] POPESCU, A.-M., AND ETZIONI, O. Extracting product features and opinions from reviews. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (2005), pp. 339–346.
- [94] QIU, G., LIU, B., BU, J., AND CHEN, C. Expanding domain sentiment lexicon through Double Propagation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence* (2009), pp. 1199–1204.
- [95] QUINLAN, J. R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.
- [96] QUINN, A. J., AND BEDERSON, B. B. Human computation: A survey and taxonomy of a growing field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), pp. 1403–1412.
- [97] QUIRK, R., GREENBAUM, S., LEECH, G., AND SVARTVIK, J. *A Comprehensive Grammar of the English Language*. Longman, 1985.
- [98] QUORA. What is average CTR in Google AdWords? <https://www.quora.com/What-is-average-CTR-in-Google-AdWords>. Accessed: 2016-09-07.
- [99] RABINER, L. R. A tutorial on Hidden Markov Models and selected applications in speech recognition. In *Readings in Speech Recognition*. Morgan Kaufmann Publishers Inc., 1990, pp. 267–296.

-
- [100] RYAN, R. M., AND DECI, E. L. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology* 25, 1 (2000), 54–67.
- [101] SAURI, R., DOMINGO, J., AND BADIA, T. The NewSoMe corpus: A unifying opinion annotation framework across genres and in multiple languages. In *Proceedings of the 9th International Conference on Language Resources and Evaluation* (2014).
- [102] SAYEED, A. B., RUSK, B., PETROV, M., NGUYEN, H. C., MEYER, T. J., AND WEINBERG, A. Crowdsourcing syntactic relatedness judgements for opinion mining in the study of information technology adoption. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities* (2011), pp. 69–77.
- [103] SCHARL, A., SABOU, M., GINDL, S., RAFELSBERGER, W., AND WEICHSELBRAUN, A. Leveraging the wisdom of the crowds for the acquisition of multilingual language resources. In *Proceedings of the 8th International Conference on Language Resources and Evaluation* (2012), pp. 379–383.
- [104] SINTSOVA, V., MUSAT, C. C., AND PU, P. Fine-grained emotion recognition in olympic tweets based on human computation. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* (2013), pp. 12–20.
- [105] SIORPAES, K., AND HEPP, M. OntoGame: Weaving the semantic web by online games. In *Proceedings of the 5th European Semantic Web Conference* (2008), pp. 751–766.
- [106] SNOW, R., O’CONNOR, B., JURAFSKY, D., AND NG, A. Y. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2008), pp. 254–263.
- [107] SOCHER, R., PERELYGIN, A., WU, J. Y., CHUANG, J., MANNING, C. D., NG, A. Y., AND POTTS, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2013), pp. 1631–1642.
- [108] SØGAARD, A., ELMING, J., AND JOHANNSEN, A. Using crowdsourcing to get representations based on regular expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2013), pp. 1476–1480.
- [109] STANFORD, U. The Stanford Parser: A statistical parser. <http://nlp.stanford.edu/software/lex-parser.shtml>. Accessed: 2016-06-15.
- [110] STONE, P. J., DUNPHY, D. C., SMITH, M. S., AND OGILVIE, D. M. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, 1966.
- [111] STOWASSER, L. Building an RPG in JavaScript with Crafty. <http://craftyjs.com/demos/tutorial/tutorial.html>. Accessed: 2016-06-15.

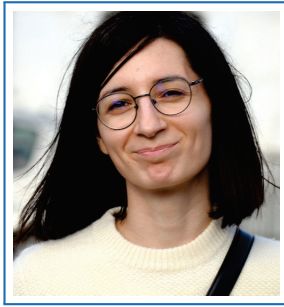
Bibliography

- [112] SU, H., DENG, J., AND FEI-FEI, L. Crowdsourcing annotations for visual object detection. In *Proceedings of the Workshops at the 26th AAAI Conference on Artificial Intelligence* (2012), pp. 40–46.
- [113] TOPRAK, C., JAKOB, N., AND GUREVYCH, I. Sentence and expression level annotation of opinions in user-generated discourse. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (2010), pp. 575–584.
- [114] TORPEY, D. An introduction to the Crafty game engine. <http://buildnewgames.com/introduction-to-crafty>. Accessed: 2016-06-15.
- [115] TURNEY, P. D. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (2002), pp. 417–424.
- [116] VINCENT, M., AND WINTERSTEIN, G. Construction et exploitation d’un corpus Français pour l’analyse de sentiment. *TALN-RÉCITAL* (2013), 764–771.
- [117] VON AHN, L. *Human Computation*. PhD thesis, Carnegie Mellon University, 2005.
- [118] VON AHN, L., AND DABBISH, L. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2004), pp. 319–326.
- [119] VON AHN, L., GINOSAR, S., KEDIA, M., LIU, R., AND BLUM, M. Improving accessibility of the web with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2006), pp. 79–82.
- [120] VON AHN, L., KEDIA, M., AND BLUM, M. Verbosity: A game for collecting common-sense facts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2006), pp. 75–78.
- [121] VON AHN, L., LIU, R., AND BLUM, M. Peekaboom: A game for locating objects in images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2006), pp. 55–64.
- [122] WACHSMUTH, H., TRENKMANN, M., STEIN, B., ENGELS, G., AND PALAKARSKA, T. A review corpus for argumentation analysis. In *Proceedings of the 15th International Conference on Computational Linguistics and Intelligent Text Processing* (2014), pp. 115–127.
- [123] WALL, M. Can we predict Oscar winners using data analytics alone? <http://www.bbc.com/news/business-35643048>, 2016. Accessed: 2016-06-13.
- [124] WANG, B., AND WANG, H. Bootstrapping both product features and opinion words from Chinese customer reviews with cross-inducing. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing* (2008), pp. 289–295.

-
- [125] WANG, S., AND MANNING, C. D. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics* (2012), pp. 90–94.
- [126] WHITEHEAD, M., AND YAEGER, L. Building a general purpose cross-domain sentiment mining model. In *Proceedings of the WRI World Congress on Computer Science and Information Engineering* (2009), pp. 472–476.
- [127] WIEBE, J., WILSON, T., AND CARDIE, C. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation* 39, 2-3 (2005), 165–210.
- [128] WIKIPEDIA. Part of speech. https://en.wikipedia.org/wiki/Part_of_speech. Accessed: 2016-06-15.
- [129] WIKIPEDIA. Syntax. <https://en.wikipedia.org/wiki/Syntax>. Accessed: 2016-06-15.
- [130] WILSON, T., WIEBE, J., AND HOFFMANN, P. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (2005), pp. 347–354.
- [131] WU, Y., AND WEN, M. Disambiguating dynamic sentiment ambiguous adjectives. In *Proceedings of the 23rd International Conference on Computational Linguistics* (2010), pp. 1191–1199.
- [132] WU, Y., ZHANG, Q., HUANG, X., AND WU, L. Phrase dependency parsing for opinion mining. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2009), pp. 1533–1541.
- [133] XIA, R., AND ZONG, C. Exploring the use of word relation features for sentiment classification. In *Proceedings of the 23rd International Conference on Computational Linguistics* (2010), pp. 1336–1344.
- [134] YANG, B., AND CARDIE, C. Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (2013), pp. 1640–1649.
- [135] YU, J., ZHA, Z.-J., WANG, M., AND CHUA, T.-S. Aspect ranking: Identifying important product aspects from online consumer reviews. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* (2011), pp. 1496–1505.
- [136] ZHANG, L., AND LIU, B. Identifying noun product features that imply opinions. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* (2011), pp. 575–580.
- [137] ZHANG, L., LIU, B., LIM, S. H., AND O’BIEN-STRAIN, E. Extracting and ranking product features in opinion documents. In *Proceedings of the 23rd International Conference on Computational Linguistics* (2010), pp. 1462–1470.

Bibliography

- [138] ZHAO, J., LIU, K., AND WANG, G. Adding redundant features for CRFs-based sentence sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2008), pp. 117–126.
- [139] ZHUANG, L., JING, F., AND ZHU, X.-Y. Movie review mining and summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management* (2006), pp. 43–50.



Marina Boia

Curriculum Vitae

Education

- 2011–2016 **PhD in Computer, Communication and Information Sciences**, *École Polytechnique Fédérale de Lausanne (EPFL)*.
- 2009–2011 **MsC in Technical Artificial Intelligence**, *Vrije Universiteit Amsterdam (VU)*, GPA – 8.8 out of 10.
- 2006–2009 **BsC in Computer Science**, *University of Bucharest*, GPA – 10 out of 10.

Experience

Teaching

- 2013–2016 **Artificial Intelligence**, *EPFL*.
Course on basic techniques in artificial intelligence. I supervised the exercise sessions, proposed exam questions, and corrected exams.
- 2012 **Analysis III**, *EPFL*.
Course on vectorial analysis. I supervised the exercise sessions.
- 2011 **Distributed Algorithms**, *VU*.
Course on concurrency concepts and various distributed algorithms. I supervised the exercise sessions.

Work

- 2015 **Software Engineer Intern**, *Google Inc., Pittsburgh*.
- 2011 **Master Thesis Intern**, *TomTom Places, Amersfoort*.
- 2008 **Computer Programmer**, *MiSyS PLC, Bucharest*.

Publications

- 2014 **Acquiring Commonsense Knowledge for Sentiment Analysis through Human Computation**, *Marina Boia, Claudiu Cristian Musat, Boi Faltings*, AAAI 2014.
- 2014 **Acquiring Commonsense Knowledge for Sentiment Analysis Using Human Computation (Poster)**, *Marina Boia, Claudiu Cristian Musat, Boi Faltings*, WWW Companion 2014.

EPFL IC IINFCOM LIA INR238 Station 14 – 1015 Lausanne, Switzerland

☎ +41 21 69 36677 • ✉ marina.boia@epfl.ch

- 2014 **Constructing Context-Aware Sentiment Lexicons with an Asynchronous Game with a Purpose**, *Marina Boia, Claudiu Cristian Musat, Boi Faltings*, CLICLING 2014.
- 2013 **A :) Is Worth a Thousand Words: How People Attach Sentiment to Emoticons and Words in Tweets**, *Marina Boia, Boi Faltings, Claudiu Cristian Musat, Pearl Pu*, SocialCom 2013.

Awards

- 2013 **Outstanding Teaching Assistant Award**, *EPFL*.
- 2011 **PhD Fellowship from the Doctoral School in Computer and Communication Science**, *EPFL*.
- 2006 **Silver Medal at the Romanian National Olympiad of Mathematics**, *Ranked 15th*.
- 2005 **Bronze Medal at the Romanian National Olympiad of Mathematics**, *Ranked 28th*.

Programming skills

Proficient Java
 Basic C++, Python, PHP, JavaScript, HTML, CSS

Languages

Mothertongue **Romanian**
 Proficient **English**
 Intermediate **French**

Interests

- Biking
- Hiking
- Reading
- Movies

