ASPECT BASED SENTIMENT ANALYSIS

Ioannis (John) Pavlopoulos

PH.D. THESIS

DEPARTMENT OF INFORMATICS
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

2014

Abstract

Aspect Based Sentiment Analysis (ABSA) systems receive as input a set of texts (e.g., product reviews or messages from social media) discussing a particular entity (e.g., a new model of a mobile phone). The systems attempt to detect the main (e.g., the most frequently discussed) aspects (features) of the entity (e.g., 'battery', 'screen') and to estimate the average sentiment of the texts per aspect (e.g., how positive or negative the opinions are on average for each aspect). Although several ABSA systems have been proposed, mostly research prototypes, there is no established task decomposition for ABSA, nore are there any established evaluation measures for the subtasks ABSA systems are required to perform.

This thesis, proposes a new task decomposition for ABSA, which contains three main subtasks: aspect term extraction, aspect term aggregation, and aspect term polarity estimation. The first subtask detects single- and multi-word terms naming aspects of the entity being discussed (e.g., 'battery', 'hard disc'), called aspect terms. The second subtask aggregates (clusters) similar aspect terms (e.g., 'price' and 'cost', but maybe also 'design' and 'color'), depending on user preferences and other restrictions (e.g., the size of the screen where the results of the ABSA system will be shown). The third subtask estimates the average sentiment per aspect term or cluster of aspect terms.

For each one of the above mentioned subtasks, benchmark datasets for different kinds of entities (e.g., laptops, restaurants) were constructed during the work of this thesis. New evaluation measures are introduced for each subtask, arguing that they

are more appropriate than previous evaluation measures. For each subtask, the thesis also proposes new methods (or improvements over previous methods), showing experimentally on the constructed benchmark datasets that the new methods (or the improved versions) are better or at least comparable to state of the art ones.

Acknowledgements

I would like to thank G. Batistatos, A. Zosakis, and G. Lampouras for their annotations in Phase A of Chapter 3, and A. Kosmopoulos, G. Lampouras, P. Malakasiotis, and I. Lourentzou for their annotations in Phase B of Chapter 3. I would also like to specially thank the following people.

Ion Androutsopoulos, my thesis supervisor, offered me valuable, consistent, and restless guidance through all the years of our co-operation. At times, it felt like he was being a PhD student, again, only to show me the way, and I thank him deeply for this. I thank my sister for being there for me, before I even think about it. I thank my parents for being my parents, and also for not being scared from what is unknown to them, and for letting it be. I also thank Aristotelis Zosakis and Makis Malakasiotis, for our many, long discussions. Many thanks to the members of the Natural Language Processing Group of AUEB's Department of Informatics, for their advice, discussions and support. I would also like to thank all the people who stood by me all these difficult years, who supported me, and lived through and with me, both the good and the bad. Lastly, I also feel I should thank all the people who wanted and tried to support me, but failed. I thank them for their efforts, but, most importantly, for letting me explore deeper sentiments and feelings.

Contents

Chapter 1

An Introduction to Aspect Based Sentiment Analysis

1.1 Subject and contribution of this thesis

Aspect Based Sentiment Analysis (ABSA) systems receive as input a set of texts (e.g., product reviews or messages from social media) discussing a particular entity (e.g., a new model of a mobile phone). The systems attempt to detect the main (e.g., the most frequently discussed) aspects (features) of the entity (e.g., 'battery', 'screen') and to estimate the average sentiment of the texts per aspect (e.g., how positive or negative the opinions are on average for each aspect). Although several ABSA systems have been proposed, mostly research prototypes (Liu, 2012), there is no established task decomposition for ABSA, nore are there any established evaluation measures for the subtasks ABSA systems are required to perform.

This thesis, proposes a new task decomposition for ABSA, which contains three main subtasks: aspect term extraction, aspect term aggregation, and aspect term polarity estimation. The first subtask detects single- and multi-word terms naming aspects of the entity being discussed (e.g., 'battery', 'hard disc'); hereafter, these terms are called

aspect terms. The second subtask aggregates (clusters) similar aspect terms (e.g., 'price' and 'cost', but maybe also 'design' and 'color'), depending on user preferences and other restrictions (e.g., the size of the screen where the results of the ABSA system will be shown). The third subtask estimates the average sentiment per aspect term or cluster of aspect terms.

For each one of the above mentioned subtasks, benchmark datasets for different kinds of entities (e.g., laptops, restaurants) were constructed during the work of this thesis. New evaluation measures are introduced for each subtask, arguing that they are more appropriate than previous evaluation measures. For each subtask, the thesis also proposes new methods (or improvements over previous methods), showing experimentally on the constructed benchmark datasets that the new methods (or the improved versions) are better or at least comparable to state of the art ones.

More specifically, for the aspect term extraction (ATE) subtask, new benchmark datasets were constructed, which have also been adopted (with some changes) by an international challenge (the ABSA Task of SEMEVAL 2014 and 2015) coorganized by the author (Pontiki et al., 2014). Also, it was shown that there is reasonable agreement between human judges (inter-annotator agreement), when they are asked to annotate aspect terms in texts. The thesis introduces new weighted variants of precision, recall, and average precision, explaining why the new variants are better than the standard ones when evaluating ATE methods. The thesis also proposes an improved form of a popular unsupervised ATE method (Hu and Liu, 2004), where an extra pruning stage that removes candidate aspect terms is added. The new pruning stage is based on recently popular methods that map words to continuous space vectors (Mikolov et al., 2013a; Mikolov et al., 2013b; Mikolov et al., 2013c). Lastly, the thesis shows experimentally, using the introduced datasets and evaluation measures, that the new improved method, with the extra pruning stage, is significantly better than the original method.

In the aspect aggregation subtask of ABSA, the thesis introduces the problem of as-

pect aggregation *at multiple granularities* and proposes decomposing the problem in two phases. In the first phase, systems attempt to fill in a similarity matrix; the value of each cell shows the semantic relatedness between two (frequent) aspect terms. In the second phase, systems use the generated similarity matrix of the first phase, along with a linkage criterion, and perform hierarchical agglomerative clustering in order to create an aspect term hierarchy; by intersecting the hierarchy at different depths, different numbers of clusters are produced, satisfying different user preferences and other restrictions (e.g., size of screen). The thesis shows experimentally, using aspect aggregation datasets constructed by the author, that the proposed decomposition leads to high inter-annotator agreement and allows re-using existing similarity measures (for the first phase) and hierarchical clustering methods (for the second phase). A novel sense pruning mechanism was also devised, which improves significantly all the existing WordNet-based similarity measures that were tested in the first phase. The experimental results show, however, that there is still large scope for improvements in the methods of the first phase. Lastly, the thesis shows that the second phase is not really affected by the linkage criterion and that it leads to near perfect results (based on human judgments) when a human-generated similarity matrix is used in the first phase. However, when the similarity matrix is generated by some system of the first phase (even the best performing system), then the results in the second phase deteriorate significantly. This shows that the second phase is in effect an almost solved problem and that future work should focus on the first phase.

For aspect term polarity estimation, the author first participated in the development of a system that estimates the sentiment (positive, negative, or neutral) of whole messages (e.g., tweets). The system is based on two classifiers (SVMs), one that detects messages carrying sentiment (positive or negative) and another one that distinguishes the positive messages from the negative ones. The use of two classifiers allows the system to deal with imbalanced sentiment classes (neutral messages can often be as many

as the positive and negative ones together). The classifiers use features based on existing sentiment lexica and sentiment scores automatically added to the entries of the lexica. The sentiment estimation system participated in international competitions (Sentiment Analysis in Twitter task of SEMEVAL 2013 and 2014) with very good results. The results also showed that the system performed better than most of the other competitors when applied to messages of a different nature than those seen during training (e.g., SMS messages instead of tweets); thus, the system has a very good generalization ability.

The message-level sentiment estimation system of the previous paragraph was then tested on aspect term polarity estimation datasets, constructed during the work of this thesis, where the goal is to estimate the sentiment for each *aspect term*, rather than for each entire message (e.g., sentence). In this case, the system of the previous paragraph was used to classify each entire message into a sentiment class (positive, negative, or neutral), and then the sentiment class of the message was also assigned to all of the aspect terms of the message, assuming that all the aspect terms of a message carry the same sentiment. Although there are messages containing aspect terms of different polarities (e.g., one positive and one negative aspect term), an analysis showed that messages of this kind are relatively rare, at least in the datasets of the experiments. Consequently, the system performs reasonably well compared to competing systems, even though it was not re-trained (unlike the competition). A new evaluation measure for the subtask of aspect term polarity estimation was also proposed, which takes into account that (i) misclassifying, for example, a positive aspect term into the negative class is a bigger mistake than misclassifying it into the neutral category; and (ii) that the end users of ABSA systems are interested mainly in frequent aspect terms; mistakes not affecting the average polarity scores of frequent aspect terms do not really matter. With the new evaluation measure, the performance of the message-level system of the previous paragraph is even better compared to its competitors. The ranking of the top

systems that participated in the corresponding subtask of an international competition (the ABSA Task of SEMEVAL 2014 and 2015) is also changed, when the new measure is used, as opposed to using the official measures of the competition. The datasets that were created for this subtask during the work of this thesis were also used (with some changes) in the same competition.

1.2 Outline of the reminder of this thesis

The reminder of this thesis is organized as follows:

- Chapter 2 discusses the work of the thesis that was devoted to aspect term extraction
- Chapter 3 discusses the work of the thesis that addressed aspect aggregation
- Chapter 4 discusses the work of the thesis that focused on message-level sentiment estimation
- Chapter 5 is concerned with aspect term polarity estimation
- Chapter 6 concludes and proposes future work

Chapter 2

Aspect Term Extraction[1]

2.1 Introduction

In this chapter, we focus on aspect term extraction (ATE). Our contribution is threefold. Firstly, we argue (Section 2.2) that previous ATE datasets are not entirely satisfactory, mostly because they contain reviews from a particular domain only (e.g., consumer electronics), or they contain reviews for very few target entities, or they do not contain annotations for aspect terms. We constructed and make publicly available three new ATE datasets with customer reviews for a much larger number of target entities from three domains (restaurants, laptops, hotels), with gold annotations of all the aspect term occurrences; we also measured inter-annotator agreement, unlike previous datasets.

Secondly, we argue (Section 2.3) that commonly used evaluation measures are also not entirely satisfactory for ATE. For example, when precision, recall, and F-measure are computed over *distinct* aspect terms (types), equal weight is assigned to more and less frequent aspect terms, whereas frequently discussed aspect terms are more important; and when precision, recall, and F-measure are computed over aspect term *occurrences* (tokens), methods that identify very few, but very frequent aspect terms may

[1] A summary of this chapter has been published (Pavlopoulos and Androutsopoulos, 2014).

appear to perform much better than they actually do. We propose weighted variants of precision and recall, which take into account the *rankings* of the distinct aspect terms that are obtained when the distinct aspect terms are ordered by their true and predicted frequencies. We also compute the average weighted precision over several weighted recall levels, in the same way that average (weighted) precision is computed over several (unweighted) recall levels (Manning and Schütze, 1999).

Thirdly, we show (Section 2.4) how the popular *unsupervised* ATE method of Hu and Liu (2004) can be extended with a pruning stage based on continuous space word vectors (Mikolov et al., 2013a; Mikolov et al., 2013b; Mikolov et al., 2013c). Using our datasets and evaluation measures, we demonstrate (Section 2.5) that the extended method performs better than the original one of Hu and Liu and a commonly used frequency-based baseline. We also report work we have performed with LDA-based methods (Blei et al., 2003b) and the reasons that led is to the decision not to consider methods of this kind further in ATE.

2.2 Datasets

We first discuss previous datasets that have been used for ATE, and we then introduce our own.

2.2.1 Previous datasets

So far, ATE methods have been evaluated mainly on customer reviews, often from the consumer electronics domain (Hu and Liu, 2004; Popescu and Etzioni, 2005; Ding et al., 2008).

The most commonly used dataset is that of Hu and Liu (2004), which contains reviews of only five particular electronic products (e.g., Nikon Coolpix 4300). Each sentence is manually annotated with aspect terms, but inter-annotator agreement has

not been reported.[2] All the sentences appear to have been selected to express clear positive or negative opinions. There are no sentences expressing conflicting opinions about aspect terms (e.g., "The screen is clear but small"), nor are there any sentences that do not express opinions about their aspect terms (e.g., "It has a 4.8-inch screen"). Hence, the dataset is not entirely representative of product reviews. By contrast, our datasets, discussed below, contain reviews from three domains, including sentences that express conflicting or no opinions about aspect terms, they concern many more target entities (not just five), and we have also measured inter-annotator agreement.

The dataset of Ganu et al. (2009), on which one of our datasets is based, is also popular. In the original dataset, each sentence is tagged with coarse aspects ('food', 'service', 'price', 'ambiance', 'anecdotes', or 'miscellaneous'). For example, "The restaurant was expensive, but the menu was great" would be tagged with the coarse aspects 'price' and 'food'. The coarse aspects, however, are not necessarily terms occurring in the sentence, and it is unclear how they were obtained. By contrast, we asked human annotators to mark the explicit aspect *terms* of each sentence, leaving the task of clustering the terms to produce coarser aspects for an aspect aggregation stage.

The 'Concept-Level Sentiment Analysis Challenge' of ESWC 2014 uses the dataset of Blitzer et al. (2007), which contains customer reviews of DVDs, books, kitchen appliances, and electronic products, with an overall sentiment score for each review. One of the challenge's tasks requires systems to extract the aspects of each sentence and a sentiment score (positive or negative) per aspect.[3] The aspects are intended to be concepts from ontologies, not simply aspect terms. The ontologies to be used, however, are not fully specified and no dataset with sentences and gold aspects is currently available.

Overall, the previous ATE datasets are not entirely satisfactory, because they contain

[2]Each aspect term occurrence is also annotated with a sentiment score. We do not discuss these scores here, since we focus on ATE. The same comment applies to the dataset of Ganu et al. (2009) and our datasets. Sentiment scores will be discussed in Chapters 4 and 5.

[3]See `http://2014.eswc-conferences.org/`.

	sentences containing n aspect term occurrences			
Domain	$n = 0$	$n \geq 1$	$n \geq 2$	total ($n \geq 0$)
Restaurants	1,590	2,120	872	3,710
Hotels	1,622	1,978	652	3,600
Laptops	1,760	1,325	416	3,085

Table 2.1: Statistics about the three aspect term extraction datasets.

	single/multi-word distinct aspect terms with n occurrences	
Domain	$n \geq 1$	$n \geq 2$
Restaurants	452/593	195/67
Hotels	262/199	120/24
Laptops	289/350	137/67

Table 2.2: More statistics about the three aspect term extraction datasets.

reviews from a particular domain only, or reviews for very few target entities, or their sentences are not entirely representative of customer reviews, or they do not contain annotations for aspect terms, or no inter-annotator agreement has been reported. To address these issues, we provide three new ATE datasets, which contain customer reviews of restaurants, hotels, and laptops, respectively.[4]

2.2.2 Our datasets

In Table 2.1, we show the number of sentences of our datasets and how many aspect term occurrences they contain. The second column ($n = 0$) shows that there are many sentences with no aspect terms. The third ($n \geq 1$) and fourth ($n \geq 2$) columns show that

[4]Our datasets are available upon request. The datasets of the ABSA task of SemEval 2014 (`http://alt.qcri.org/semeval2014/task4/`) are based on our datasets.

most sentences contain exactly one aspect term occurrence.

In Table 2.2, we see the total number of single- and multi-word distinct aspect terms (column $n \geq 1$), and also the number of single- and multi-word distinct aspect terms that occur more than once in each dataset (column $n \geq 2$). If we consider all of the distinct aspect terms (column $n \geq 1$), the multi-word aspect terms are more than the single-word aspect terms in the restaurant and laptop reviews, but not in the hotel reviews. However, if we consider only distinct aspect terms that occur more than once in each dataset (column $n \geq 2$), then the single-word distinct aspect terms are always more. This means that many multi-word aspect terms occur only once in each dataset; and we do not really care about aspect terms that occur only once in a dataset. In our experiments, we ignore aspect terms that occur only once.

Our restaurants dataset contains 3,710 English sentences (Table 2.1) from the reviews of Ganu et al. (2009).[5] We asked human annotators to tag the aspect terms of each sentence. In "The *dessert* was divine", for example, the annotators would tag the aspect term 'dessert'. In a sentence like "The restaurant was expensive, but the *menu* was great", the annotators were instructed to tag only the explicitly mentioned aspect term 'menu'. The sentence also refers to the prices, and a possibility would be to add 'price' as an *implicit aspect term*, but we do not consider implicit aspect terms in this thesis. We used nine annotators for the restaurant reviews. The annotators were graduate Computer Science students, fluent in English, but not native English speakers. Each sentence was processed by a single annotator, and each annotator processed approximately the same number of sentences.

Our hotels dataset contains 3,600 English sentences (Table 2.1) from online customer reviews of 30 hotels. We used three annotators with the same background as in the restaurants dataset. Again, each sentence was processed by a single annotator,

[5]The original dataset of Ganu et al. contains 3,400 sentences, but some of the sentences had not been properly split.

and each annotator processed approximately the same number of sentences. Our laptops dataset contains 3,085 English sentences of 394 online customer reviews. A single annotator (one of the authors) was used in the laptops dataset.

To measure inter-annotator agreement, we used a sample of 75 restaurant, 75 laptop, and 100 hotel sentences. Each sentence was processed by two (for restaurants and laptops) or three (for hotels) annotators, other than the annotators used previously. For each sentence s_i, the inter-annotator agreement was measured as the Dice coefficient $D_i = 2 \cdot \frac{|A_i \cap B_i|}{|A_i| + |B_i|}$, where A_i, B_i are the sets of aspect term occurrences tagged by the two annotators, respectively, and $|S|$ denotes the cardinality of a set S; for hotels, we use the mean pairwise D_i of the three annotators.[6] The overall inter-annotator agreement D was taken to be the average D_i of the sentences of each sample. We, thus, obtained $D = 0.72, 0.70, 0.69$, for restaurants, hotels, and laptops, respectively, which indicate reasonably high inter-annotator agreement.

2.2.3 Single and multi-word aspect terms

ABSA systems use ATE methods ultimately to obtain the m most prominent (frequently discussed) distinct aspect terms of the target entity, for different values of m.[7] In a system like the one of Fig. 2.1, for example, if we ignore aspect aggregation, each row will report the average sentiment score of a single frequent distinct aspect term, and m will be the number of rows, which may depend on the display size or user preferences.

Figure 2.2 shows the percentage of distinct multi-word aspect terms among the m most frequent distinct aspect terms, for different values of m, in our three datasets and the electronics dataset of Hu and Liu (2004). There are many more single-word distinct

[6] Cohen's Kappa cannot be used here, because the annotators may tag any word sequence of any sentence, which leads to a very large set of categories. A similar problem was reported by Kobayashi et al. (2007).

[7] A more general definition of prominence might also consider the average sentiment score of each distinct aspect term.

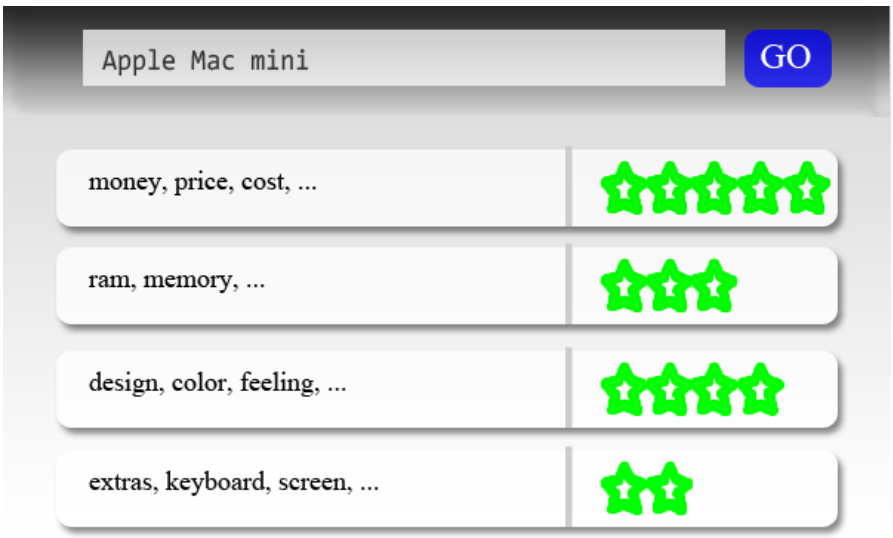


Figure 2.1: Automatically extracted prominent aspects (shown as clusters of aspect terms) and average aspect sentiment scores of a target entity.

aspect terms than multi-word distinct aspect terms, especially in the restaurant and hotel reviews. In the electronics and laptops datasets, the percentage of multi-word distinct aspect terms (e.g., 'hard disk') is higher, but most of the distinct aspect terms are still single-word, especially for small values of m. By contrast, many ATE methods (Hu and Liu, 2004; Popescu and Etzioni, 2005; Wei et al., 2010) devote much of their processing to identifying multi-word aspect terms, which may be the right priority in hotel and restaurant reviews. Figure 2.2 also shows that the ratio of multi-word to single-word distinct aspect terms may be significantly different across domains, which is an example of why it is important to consider reviews from multiple domains.

2.3 Evaluation measures

We now discuss previous ATE evaluation measures, also introducing our own.

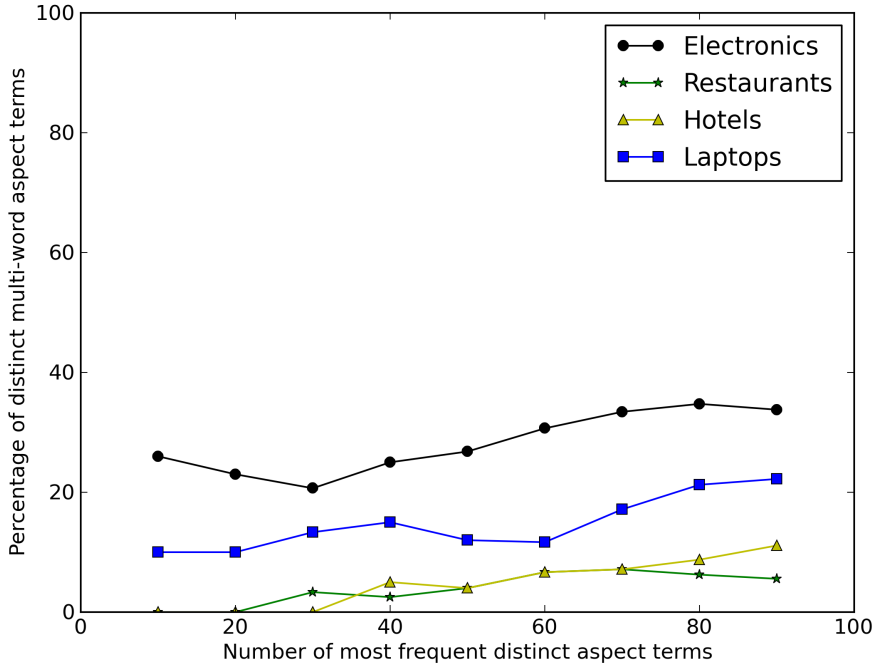


Figure 2.2: Percentage of (distinct) multi-word aspect terms among the most frequent aspect terms.

2.3.1 Precision, Recall, F-measure

ATE methods are usually evaluated using precision, recall, and F-measure (Hu and Liu, 2004; Popescu and Etzioni, 2005; Kim and Hovy, 2006; Wei et al., 2010; Moghaddam and Ester, 2010; Bagheri et al., 2013), but it is often unclear if these measures are applied to *distinct* aspect terms (types, no duplicates) or aspect term *occurrences* (tokens).

In the former case, each method is expected to return a set A of distinct aspect terms, to be compared to the set G of distinct aspect terms the human annotators identified in the texts. TP (true positives) is $|A \cap G|$, FP (false positives) is $|A \setminus G|$, FN (false

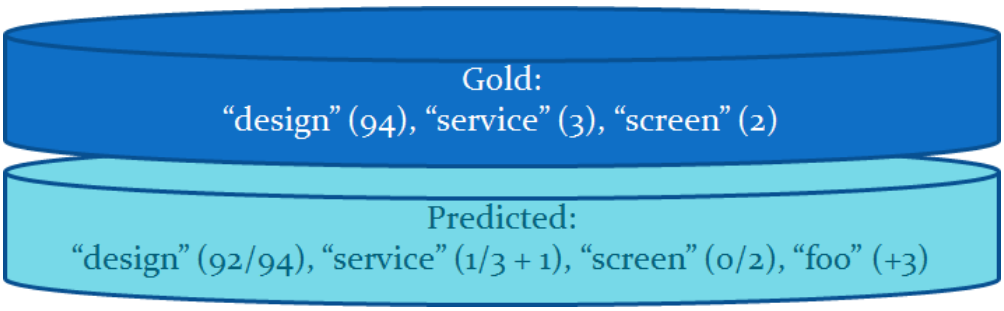


Figure 2.3: Distinct aspect terms and aspect term occurrences tagged by human annotators (Gold) and a system (predicted).

negatives) is $|G \setminus A|$, and precision (P), recall (R), $F = \frac{2 \cdot P \cdot R}{P+R}$ are defined as usually:

$$P = \frac{TP}{TP + FP}, \; R = \frac{TP}{TP + FN} \tag{2.1}$$

This way, however, precision, recall, and F-measure assign the same importance to all the distinct aspect terms, whereas missing, for example, a more frequent (more frequently discussed) distinct aspect term should probably be penalized more heavily than missing a less frequent one.

Assume, for example, that the human annotators have tagged 94 occurrences of the word 'design' as aspect terms, three occurrences of the word 'service', and two occurrences of 'screen' (Fig. 2.3). Suppose now that a system tagged correctly as aspect terms 92 of the 94 occurrences of 'design' that the humans had tagged, one of the three correct occurrences of 'service', plus one wrong occurrence of 'service' that the humans did not consider an aspect term. Also, the system did not tag any occurrences of 'screen' and it also tagged three occurrences of the word 'foo' that the humans never tagged. When precision, recall, and F-measure are computed over distinct aspect terms, term frequency plays no role at all and the penalty or reward for missing entirely or finding a rare aspect term, like 'screen', is the same as the penalty or reward for missing or finding a very frequent aspect term, like 'design', which does not seem right.

When precision, recall, and F-measure are applied to aspect term *occurrences* (Liu

et al., 2005), *TP* is the number of aspect term occurrences tagged (each term occurrence) both by the method being evaluated and the human annotators, *FP* is the number of aspect term occurrences tagged by the method but not the human annotators, and *FN* is the number of aspect term occurrences tagged by the human annotators but not the method. The three measures are then defined as above. They now assign more importance to frequently occurring distinct aspect terms, but they can produce misleadingly high scores when only a few, but very frequent distinct aspect terms are handled correctly. Furthermore, the occurrence-based definitions do not take into account that missing several aspect term occurrences or wrongly tagging expressions as aspect term occurrences may not actually matter, as long as the m most frequent distinct aspect terms can be correctly reported.

Returning to our example of Fig. 2.3, when precision, recall, and F-measure are computed over aspect term occurrences, all three scores appear to be very high, mostly because the system performs well with the occurrences of 'design', which is a very frequent aspect term, even though it performs poorly with all the other aspect terms. This also does not seem right. In the case of distinct aspect terms, precision and recall do not consider term frequencies at all, and in the case of aspect term occurrences the two measures are too sensitive to high-frequency terms. These problems also affect F-measure, which is the harmonic mean of precision and recall.

2.3.2 Weighted precision, recall, AWP

What the previous definitions of precision and recall miss is that in practice ABSA systems use ATE methods ultimately to obtain the m most frequent distinct aspect terms, for a range of m values. Let A_m and G_m be the *lists* that contain the m most frequent distinct aspect terms, ordered by their predicted and true frequencies, respectively; the predicted and true frequencies are computed by examining how frequently the ATE method or the human annotators, respectively, tagged occurrences of each distinct aspect term. Dif-

ferences between the predicted and true frequencies do not matter, as long as $A_m = G_m$, for every m in a range of m values we care for. Not including in A_m a term of G_m should be penalized more or less heavily, depending on whether the term's true frequency was high or low, respectively. Furthermore, including in A_m a term not in G_m should be penalized more or less heavily, depending on whether the term was placed towards the beginning or the end of A_m, i.e., depending on the prominence that was assigned to the term.

To address the issues discussed above, we introduce weighted variants of precision and recall. For each ATE method, we now compute a single *list* $A = \langle a_1, \ldots, a_{|A|} \rangle$ of distinct aspect terms identified by the method, ordered by decreasing predicted frequency. For every m value (number of most frequent distinct aspect terms to show), the method is treated as having returned the sub-list A_m with the first m elements of A. Similarly, we now take $G = \langle g_1, \ldots, g_{|G|} \rangle$ to be the *list* of the distinct aspect terms that the human annotators tagged, ordered by decreasing true frequency.[8] We define weighted precision (WP_m) and weighted recall (WR_m) as in Eq. 2.2–2.3. The notation $1\{\kappa\}$ denotes 1 if condition κ holds, and 0 otherwise. By $r(a_i)$ we denote the ranking of the returned term a_i in G, i.e., if $a_i = g_j$, then $r(a_i) = j$; if $a_i \notin G$, then $r(a_i)$ is an arbitrary positive integer.

$$WP_m = \frac{\sum_{i=1}^{m} \frac{1}{i} \cdot 1\{a_i \in G\}}{\sum_{i=1}^{m} \frac{1}{i}} \tag{2.2}$$

$$WR_m = \frac{\sum_{i=1}^{m} \frac{1}{r(a_i)} \cdot 1\{a_i \in G\}}{\sum_{j=1}^{|G|} \frac{1}{j}} \tag{2.3}$$

The numerator of the definition of WR_m (Eq. 2.3) counts how many terms of G (gold distinct aspect terms) the method returned in A_m, but weighting each term by its inverse ranking $\frac{1}{r(a_i)}$, i.e., assigning more importance to terms the human annotators tagged more frequently. The denominator of Eq. 2.3 sums the weights of all the terms of G; in unweighted recall applied to distinct aspect terms, where all the terms of G have the

[8] In our experiments, we exclude from G aspect terms tagged by the annotators only once.

same weight, the denominator would be $|G| = TP + FN$ (Eq. 2.1). The numerator of WP_m (Eq. 2.3) again counts how many gold aspect terms the method returned in A_m, but weighting each returned term a_i by its inverse ranking $\frac{1}{i}$ in A_m, to reward methods that return more gold aspect terms towards the beginning of A_m. The denominator of Eq. 2.2 sums the weights of all the terms of A_m; in unweighted precision applied to distinct aspect terms, the denominator would be $|A_m| = TP + FN$ (Eq. 2.1).

In effect, weighted recall is the same as simple recall computed over distinct aspect terms, but it considers only the top m elements of the A list, and it assigns more importance to distinct aspect terms that the human annotators tagged more frequently. Similarly, weighted precision is the same as simple precision computed over distinct aspect terms, but it considers only the top m elements of the A list, and it assigns more importance to correct distinct aspect terms placed towards the beginning of the A_m list.

Returning to our example of Fig. 2.3, for $m = 3$, $A_m = \langle design, foo, service \rangle$, $G = \langle design, service, screen \rangle$ and $WP_m = 3$ and $WR_m = 3$ would be:

$$WP_m = \frac{\frac{1}{1} + 0 + \frac{1}{3}}{\frac{1}{1} + \frac{1}{2} + \frac{1}{3}} = 0.73 \quad (2.4)$$

$$WR_m = \frac{\frac{1}{1} + 0 + \frac{1}{2}}{\frac{1}{1} + \frac{1}{2} + \frac{1}{3}} = 0.82 \quad (2.5)$$

We plot weighted precision-recall curves by computing WP_m, WR_m pairs for different values of m, as in Fig. 2.4 below (page 19).[9] The higher the curve of a method, the better the method. We also compute the average (interpolated) weighted precision (AWP) of each method over 11 recall levels:

$$AWP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} WP_{int}(r)$$

$$WP_{int}(r) = \max_{m \in \{1, \ldots, |A|\}, WR_m \geq r} WP_m$$

[9] With supervised methods, we perform a 10-fold cross-validation for each m, and we macro-average WP_m, WR_m over the folds. We provide our datasets partitioned in folds.

AWP is similar to average (interpolated) precision (*AP*), which is commonly used to summarize the tradeoff between (unweighted) precision and recall (Manning et al., 2008).

2.3.3 Other related measures

Yu at al. (2011a) used *nDCG@m* (Järvelin and Kekäläinen, 2002; Sakai, 2004; Manning et al., 2008), defined below, to evaluate each list of m distinct aspect terms returned by an ATE method.

$$nDCG@m = \frac{1}{Z} \sum_{i=1}^{m} \frac{2^{t(i)} - 1}{\log_2(1+i)}$$

Z is a normalization factor to ensure that a perfect ranking gets $nDCG@m = 1$, and $t(i)$ is a reward function for a term placed at position i of the returned list. In the work of Yu et al., $t(i) = 1$ if the term at position i is not important (as judged by a human), $t(i) = 2$ if the term is 'ordinary', and $t(i) = 3$ if it is important. The logarithm is used to reduce the reward for distinct aspect terms placed at lower positions of the returned list.

The *nDCG@m* measure is well known in ranking systems (e.g., search engines) and it is similar to our weighted precision (WP_m). The denominator or Eq. 2.2 corresponds to the normalization factor Z of *nDCG@m*; the $\frac{1}{i}$ factor of the numerator of Eq. 2.2 corresponds to the $\frac{1}{\log_2(1+i)}$ degradation factor of *nDCG@m*; and the $1\{a_i \in G\}$ factor of Eq. 2.2 is a binary reward function, corresponding to the $2^{t(i)} - 1$ factor of *nDCG@m*.

The main difference from *nDCG@m* is that WP_m uses a degradation factor $\frac{1}{i}$ that is inversely proportional to the ranking of the returned term a_i in the returned list A_m, whereas *nDCG@m* uses a logarithmic factor $\frac{1}{\log_2(1+i)}$, which reduces less sharply the reward for distinct aspect terms returned at lower positions in A_m. We believe that the degradation factor of WP_m is more appropriate for ABSA, because most users would in practice wish to view sentiment scores for only a few (e.g., $m \leq 10$) frequent distinct aspect terms, whereas in search engines users are more likely to examine more of the highly-ranked returned items. It is possible, however, to use a logarithmic degradation

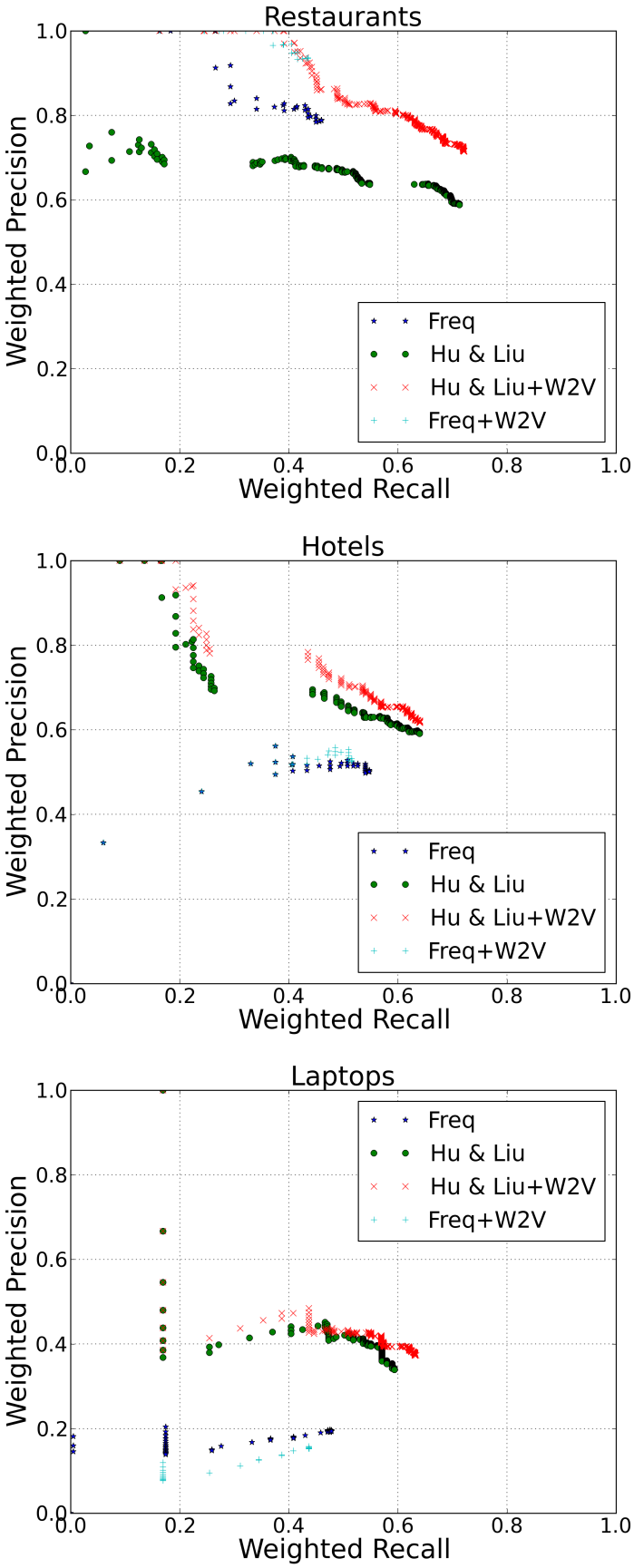


Figure 2.4: Weighted precision – weighted recall curves for our three datasets.

factor in WP_m, as in *nDCG@m*.

Another difference is that we use a binary reward factor $1\{a_i \in G\}$ in WP_m, instead of the $2^{t(i)} - 1$ factor of *nDCG@m* that has three possibly values in the work of Yu at al. (2011a). We use a binary reward factor, because preliminary experiments we conducted indicated that multiple relevance levels (e.g., not an aspect term, aspect term but unimportant, important aspect term) confused the annotators and led to lower inter-annotator agreement. The *nDCG@m* measure can also be used with a binary reward factor; the possible values $t(i)$ would then be 0 and 1, as in the case of the $1\{a_i \in G\}$ factor of WP_m.

With a binary reward factor, *nDCG@m* in effect measures the ratio of correct (distinct) aspect terms to the terms returned, assigning more weight to correct aspect terms placed closer the top of the returned list, like WP_m. The *nDCG@m* measure, however, does not provide any indication of how many of the gold distinct aspect terms have been returned. By contrast, we also measure weighted recall (Eq. 2.3), which examines how many of the (distinct) gold aspect terms have been returned in A_m, also assigning more weight to the gold aspect terms the human annotators tagged more frequently. We also compute the average weighted precision (*AWP*), which is a combination of WP_m and WR_m, for a range of m values.

2.4 Aspect term extraction methods

We implemented and evaluated four ATE methods: (i) a popular baseline (dubbed FREQ) that returns the most frequent distinct nouns and noun phrases, (ii) the well-known method of Hu and Liu (2004), which adds to the baseline pruning mechanisms and steps that detect more aspect terms (dubbed H&L), (iii) an extension of the previous method (dubbed H&L+W2V) with an extra pruning step we devised that uses the recently popular continuous space word vectors (Mikolov et al., 2013c), and (iv) a similar

extension of FREQ (dubbed FREQ+W2V). All four methods are *unsupervised*, which is particularly important for ABSA systems intended to be used across domains with minimal changes. They return directly a list A of distinct aspect terms ordered by decreasing predicted frequency, rather than tagging aspect term occurrences. We note, however, that our evaluation measures (Section 2.3.2) can also be used with ATE methods that tag aspect term occurrences, by computing the A list from the occurrences tagged by the methods, before applying the evaluation measures

2.4.1 The FREQ baseline

The FREQ baseline returns the most frequent (distinct) nouns and noun phrases of the reviews in each dataset (restaurants, hotels, laptops), ordered by decreasing sentence frequency (how many sentences contain the noun or noun phrase).[10] This is a reasonably effective and popular baseline (Hu and Liu, 2004; Wei et al., 2010; Liu, 2012).

2.4.2 The H&L method

The method of Hu and Liu (2004), dubbed H&L, first extracts all the distinct nouns and noun phrases (excluding determiners) from the reviews of each dataset (lines 3–6 of Algorithm 1) and considers them candidate distinct aspect terms.[11] It then forms longer candidate distinct aspect terms by concatenating pairs and triples of candidate aspect terms occurring in the same sentence, in the order they appear in the sentence (lines 7–11). For example, if 'battery life' and 'screen' occur in the same sentence (in this order), then 'battery life screen' will also become a candidate distinct aspect term.

The resulting candidate distinct aspect terms are ordered by decreasing *p-support*

[10] We use the default POS tagger of NLTK, and the chunker of NLTK trained on the Treebank corpus; see `http://nltk.org/`. We convert all words to lower-case.

[11] Some details of the work of Hu and Liu (2004) were not entirely clear to us. The discussion here and our implementation reflect our understanding.

(lines 12–15). The *p-support* of a candidate distinct aspect term t is the number of sentences that contain t, excluding sentences that contain another candidate distinct aspect term t' that subsumes t. For example, if both 'battery life' and 'battery' are candidate distinct aspect terms, a sentence like "The battery life was good" is counted in the *p-support* of 'battery life', but not in the *p-support* of 'battery'.

The method then tries to correct itself by pruning wrong candidate distinct aspect terms and detecting additional candidates. Firstly, it discards multi-word distinct aspect terms that appear in 'non-compact' form in more than one sentences (lines 16–23). A multi-word term t appears in non-compact form in a sentence if there are more than three other words (not words of t) between any two of the words of t in the sentence. For example, the candidate distinct aspect term 'battery life screen' appears in non-compact form in "battery life is way better than screen". Secondly, if the *p-support* of a candidate distinct aspect term t is smaller than 3 and t is subsumed by another candidate distinct aspect term t', then t is discarded (lines 21–23).

Subsequently, a set of 'opinion adjectives' is formed; for each sentence and each candidate distinct aspect term t that occurs in the sentence, the closest to t adjective of the sentence (if there is one) is added to the set of opinion adjectives (lines 25-27). The sentences are then re-scanned; if a sentence does not contain any candidate aspect term, but contains an opinion adjective, then the nearest noun to the opinion adjective is added to the candidate distinct aspect terms (lines 28–31). The remaining candidate distinct aspect terms are returned, ordered by decreasing *p-support*.

2.4.3 The H&L+W2V method

We extended H&L by including an additional pruning step that uses continuous vector space representations of words (Mikolov et al., 2013a; Mikolov et al., 2013b; Mikolov et al., 2013c). The vector representations of the words can be produced, for example, by training a language model to predict the following word, or by training a model to

Algorithm 1 The method of Hu and Liu

Require: sentences: a list of sentences

```
terms = new Set(String)
psupport = new Map(String, int)
for s in sentences do
    nouns = POSTagger(s).getNouns()
    nps = Chunker(s).getNPChunks()
    terms.add(nouns ∪ nps)
for s in sentences do
    for t1, t2 in terms s.t. t1, t2 in s ∧ s.index(t1)<s.index(t2) do
        terms.add(t1 + " " + t2)
    for t1, t2, t3 in s.t. t1, t2, t3 in s ∧ s.index(t1)<s.index(t2)<s.index(t3) do
        terms.add(t1 + " " + t2 + " " + t3)
for s in sentences do
    for t: t in terms ∧ t in s do
        if ¬∃ t': t' in terms ∧ t' in s ∧ t in t' then
            psupport[term] += 1
nonCompact = new Map(String, int)
for t in terms do
    for s in sentences do
        if maxPairDistance(t.words())>3 then
            nonCompact[t] += 1
for t in terms do
    if nonCompact[t]>1 ∨ (∃ t': t' in terms ∧ t in t' ∧ psupport[t]<3) then
        terms.remove(t)
adjs = new Set(String)
for s in sentences do
    if ∃ t: t in terms ∧ t in s then
        adjs.add(POSTagger(s).getNearestAdj(t))
for s in sentences do
    if ¬∃ t: t in terms ∧ t in s ∧ ∃ a: a in adjs ∧ a in s then
        t = POSTagger(s).getNearestNoun(adjs)
        terms.add(t)
return psupport.keysSortedByValue()
```

predict the current word given the surrounding words (Mikolov et al., 2013a). In all cases, each word of the vocabulary is represented as a dense vector of a continuous vector space and the word vectors are treated as latent variables to be learned during training, consult the work of Mikolov et al. for more details. We used the English Wikipedia as the training corpus to obtain word vectors, with 200 latent features per vector. Vectors for short phrases, in our case candidate multi-word aspect terms, are produced in a similar manner.[12]

Our additional pruning stage is invoked immediately immediately after line 6 of Algorithm 1. It uses the ten most frequent candidate distinct aspect terms that are available up to that point (frequency taken to be the number of sentences that contain each candidate) and computes the centroid of their vectors, dubbed the *domain centroid*. Similarly, it computes the centroid of the 20 most frequent words of the Brown Corpus (news category), excluding stop-words and words shorter than three characters; this is the *common language centroid*.[13] Any candidate distinct aspect term whose vector is closer to the common language centroid than the domain centroid is discarded, the intuition being that the candidate names a very general concept, rather than a domain-specific aspect.[14] We use cosine similarity to compute distances. Vectors obtained from Wikipedia are used in all cases (even when computing the centroid of the most frequent Brown words).

To showcase the insight of our pruning step, Table 2.3 shows the five words from the English Wikipedia whose vectors are closest to the common language centroid and to three domain centroids. The words closest to the common language centroid are

[12] We use WORD2VEC, available at `https://code.google.com/p/word2vec/`, with a continuous bag of words model, default parameters, the first billion characters of the English Wikipedia, and the pre-processing of `http://mattmahoney.net/dc/textdata.html`.

[13] We used more words in the case of the *common language centroid* to reflect more general concepts.

[14] WORD2VEC does not produce vectors for phrases longer than two words; thus, our pruning mechanism never discards candidate aspect terms of more than two words.

Centroid	Closest Wikipedia words
Com. lang.	only, however, so, way, because
Restaurants	meal, meals, breakfast, wingstreet, snacks
Hotels	restaurant, guests, residence, bed, hotels
Laptops	gameport, hardware, hd floppy, pcs, apple macintosh

Table 2.3: Wikipedia words closest to the common language centroid and to three domain centroids.

common words, whereas words closest to the domain centroids name domain-specific concepts that are more likely to be aspect terms.

2.4.4 The FREQ+W2V method

As with H&L+W2V, we extended FREQ by adding our pruning step that uses the continuous space word (and phrase) vectors. Again, we produced one common language and three domain centroids, as before. Candidate distinct aspect terms whose vector was closer to the common language centroid than the domain centroid were discarded.

2.4.5 LDA-based methods

Latent topic models, mostly based on Latent Dirichlet Allocation (LDA) (Blei et al., 2003b), have also been used in ways that are related to ATE (Mei et al., 2007; Titov and McDonald, 2008b; Brody and Elhadad, 2010; Zhao et al., 2010; Jo and Oh, 2011). Roughly speaking, an LDA model assumes that each document d of $|d|$ words $w_1, \ldots, w_{|d|}$ is generated by iteratively (for $r = 1, \ldots, |d|$) selecting a topic t_r from a document-specific multinomial distribution $P(t|d)$ over T topics, and then selecting a word w_r from a topic-specific multinomial distribution $P(w|t)$ over the vocabulary.[15] An LDA

[15]The document-specific parameters of the first multinomial distribution are drawn from a Dirichlet distribution.

model can be trained on a corpus in an unsupervised manner to estimate the parameters of the distributions it involves.[16]

In order to test the applicability of latent topic models to ATE alone, we designed two LDA-based systems and we evaluated them on our ATE datasets. In preliminary experiments, however, the performance of the two LDA-based methods was significantly lower than the performance of the other ATE methods we considered and, thus, the LDA-based methods were excluded from further analysis. Below, we describe the two LDA-based methods we considered.

2.4.5.1 LDA+rel baseline

In our first LDA-based method, dubbed LDA+rel, we treat each sentence as a separate document d. We train an LDA model (in an unsupervised manner) on the set of sentences that constitute the input to ATE.[17] We then collect the $\frac{m}{T}$ nouns w with the highest $Rel_t(w)$ scores, defined below, from each topic t, and we return them as the output of LDA+rel, ordering them by decreasing $Rel_t(w)$ and removing duplicates.

$$Rel_t(w) = \log \frac{P(w|t)}{P(w)}$$

$Rel_t(w)$ is based on the work of Mei et al. (2007). It is intended to assess the relevance of word w to topic t. $Rel_t(w)$ prefers the most frequent words (in our case, nouns) of each topic (high $P(w|t)$), but it penalizes words that are frequent in the entire set of input sentences (high $P(w)$). $P(w)$ is the probability of encountering w in a subjective

[16] We use the LDA implementation found at `http://www.arbylon.net/projects/`. We use the default settings, i.e., we use Gibbs Sampling, 3000 iterations, 200 burn-in, 100 thin-interval, 10 sample-lag, and we set $\alpha = \beta = 0.1$.

[17] In these preliminary experiments, we also experimented with using only subjective sentences (i.e., discarding sentences carrying neutral sentiment or no sentiment at all) as input, but we there was no difference in the results.

sentence; we estimate it as:

$$P(w) = \sum_{t=1}^{T} P(w|t) \cdot P(t)$$

We estimate $P(t)$ as follows, where $|D|$ is the number of input sentences. $P(w|t)$ and $P(t|d)$ are learnt during the training of the LDA model.

$$P(t) = \frac{1}{|D|} \cdot \sum_{d=1}^{|D|} P(t|d)$$

To select the value of T (number of topics), we pick the 10 most frequent nouns (highest $P(w|t)$) of each topic, we plot the average $Rel_t(w)$ score of the $10 \cdot T$ nouns, as in Fig. 2.5, and we select the T that leads to the highest average $Rel_t(w)$. In other words, we require the 10 top most frequent nouns of each topic to be highly relevant to the topic, intuitively to constitute a good description of the topic. Figure 2.5 shows that for restaurant and laptop reviews, this process leads to selecting between 12 and 16 topics.[18] This range is consistent with the T selected by Brody and Elhadad (2010) via more complex cluster validation techniques on the same restaurant reviews; other, more elaborate methods to select T have also been proposed (Blei et al., 2003a). Each experiment of Fig. 2.5 was repeated 6 times, and the error bars correspond to 95% confidence intervals.

2.4.5.2 LDA+PMI baseline

This method is based on the work of Brody and Elhadad (2010). Following their work, a cluster validation approach is employed to find the optimum number of LDA topics. For each topic t, a score based on (normalized) Pointwise Mutual Information (PMI), defined below, is used to rank the nouns w that are strongly associated with t. $P(w)$ and $P(t)$ are estimated as above, and $P(w,t) = P(w|t) \cdot P(t)$.

$$\text{PMI}(w,t) = P(w,t) \cdot \log \frac{P(w,t)}{P(w) \cdot P(t)}$$

[18] These experiments were not performed on the hotels dataset, which was constructed later.

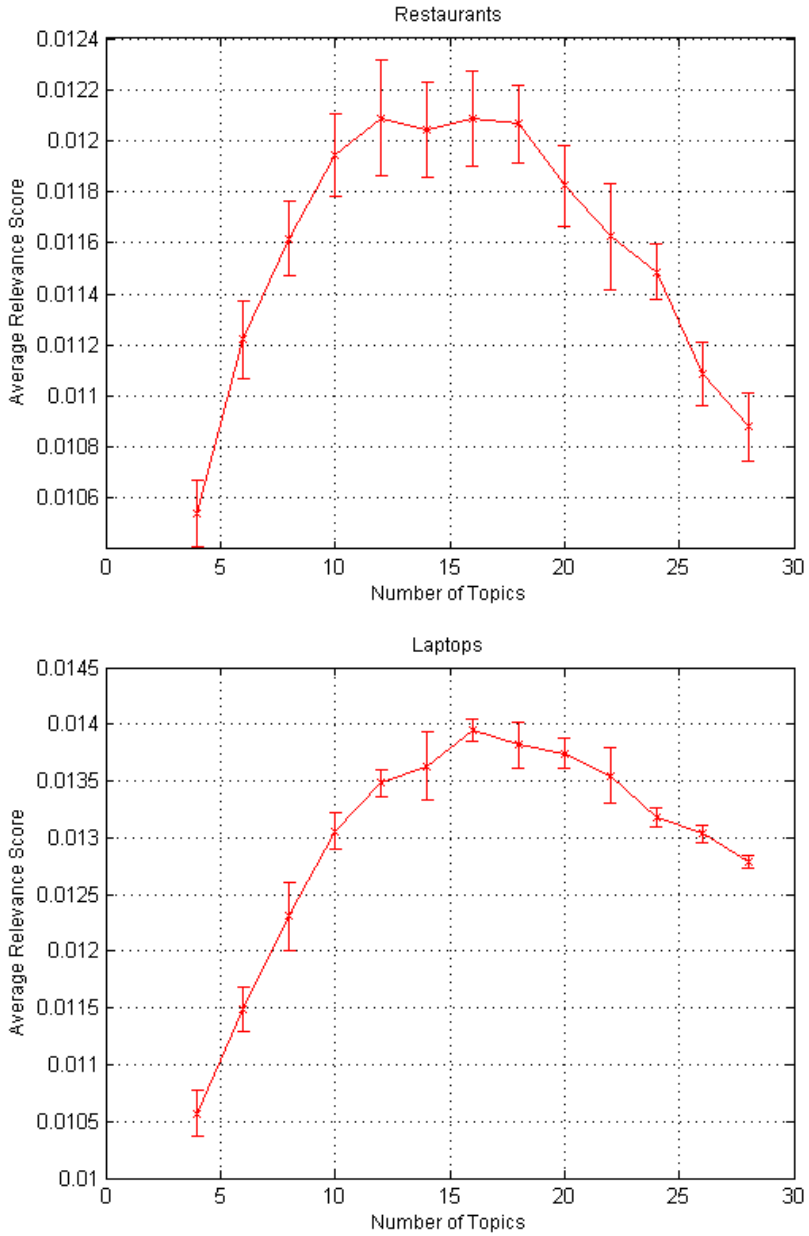


Figure 2.5: Selecting the number of topics of the LDA+rel aspect extraction baseline.

2.5 Experimental results

Table 2.4 shows the *AWP* scores of the ATE methods we considered, including the two LDA-based methods. The LDA-based methods clearly perform poorly and we do not consider them any further. All the other four methods perform better on the restaurants dataset compared to the other two datasets. At the other extreme, the laptops dataset seems to be the most difficult one; this is due to the fact that it contains many frequent nouns and noun phrases that are not aspect terms; it also contains more multi-word aspect terms (Fig. 2.2).

In all three domains, H&L performs much better than FREQ and our additional pruning (W2V) improves H&L. By contrast, FREQ benefits from W2V in the restaurant reviews (but to a smaller degree than H&L), it benefits only marginally in the hotel reviews, and in the laptop reviews FREQ+W2V performs worse than FREQ. By analysing the results, we observed that the list of candidate (distinct) aspect terms that FREQ produces already misses many aspect terms in the hotel and laptop datasets; hence, W2V, which can only prune aspect terms, cannot improve the results much, and in the case of laptops W2V has a negative effect, because it prunes several correct candidate aspect terms. All the differences between *AWP* scores (of the first four methods of Table 2.4) on the same dataset are statistically significant; we use stratified approximate randomization, which indicates $p \leq 0.01$ in all cases.[19]

Figure 2.4 (page 19) shows the weighted precision and weighted recall curves of the four methods (excluding the LDA-based methods). In the restaurants dataset, our pruning improves the weighted precision of both H&L and FREQ; by contrast, it does not improve weighted recall, since it can only prune candidate aspect terms. The maximum weighted precision of FREQ+W2V is almost as good as that of H&L+W2V, but H&L+W2V (and H&L) reach much higher weighted recall scores. In the hotel reviews, W2V again improves the weighted precision of both H&L and FREQ, but to a smaller

[19] See `http://masanjin.net/sigtest.pdf`.

Method	Restaurants	Hotels	Laptops
FREQ	43.40	30.11	9.09
FREQ+W2V	45.17	30.54	7.18
H&L	52.23	49.73	34.34
H&L+W2V	66.80	53.37	38.93
LDA+rel	0.02	-	0.03
LDA+PMI	0.07	-	0.04

Table 2.4: Average weighted precision results (%).

extent; again W2V does not improve weighted recall; also, H&L and H&L+W2V again reach higher weighted recall scores, compared to FREQ+W2V (and FREQ). In the laptop reviews, W2V marginally improves the weighted precision of H&L, but it lowers the weighted precision of FREQ; again H&L and H&L+W2V reach higher weighted recall scores. Overall, Fig. 2.4 confirms that H&L+W2V is the best method among the four tested (again, excluding the two LDA-based methods).

2.6 Conclusions

We constructed and made publicly available three new ATE datasets from three domains. We also introduced weighted variants of precision, recall, and average precision, arguing that they are more appropriate for ATE. Finally, we discussed how a popular unsupervised ATE method (H&L) can be improved by adding a new pruning mechanism that uses continuous space vector representations of words and phrases. Using our datasets and evaluation measures, we showed that the improved method performs clearly better than the original one, also outperforming a simpler frequency-based baseline with or without our pruning. We also designed two LDA-based ATE methods, but preliminary experiments indicated that these methods performed very poorly in ATE compared to

the other methods we considered and, hence, the LDA-based methods were excluded from further analysis for the purposes of ATE.

Chapter 3

Multi-Granular Aspect Aggregation[1]

3.1 Introduction

In this chapter we focus on aspect aggregation. Recall that aspect aggregation is needed to avoid reporting separate sentiment scores for aspect terms that are very similar. In Fig. 2.1 (page 12), for example, showing separate lines for 'money', 'price', and 'cost' would be confusing. The extent to which aspect terms should be aggregated, however, also depends on the available space and user preferences. On devices with smaller screens, it may be desirable to aggregate aspect terms that are similar, though not necessarily near-synonyms (e.g., 'design', 'color', 'feeling') to show fewer lines (Fig. 2.1), but finer aspects may be preferable on larger screens. Users may also wish to adjust the granularity of aspects, e.g., by stretching or narrowing the height of Fig. 2.1 on a smartphone to view more or fewer lines. Hence, aspect aggregation should be able to produce groups of aspect terms for *multiple granularities*. We assume that the aggregated aspects are displayed as lists of terms, as in Fig. 2.1. We make no effort to order (e.g., by frequency) the terms in each list, nor do we attempt to label each aggregated aspect (list of aspect terms) with a single (more general) term, leaving such tasks for

[1] A summary of this chapter has been published (Pavlopoulos and Androutsopoulos, 2014)

future work.

ABSA systems usually group synonymous (or near-synonymous) aspect terms (Liu, 2012). Aggregating only synonyms (or near-synonyms), however, does not allow users to select the desirable aspect granularity, and ignores the hierarchical relations between aspect terms. For example, 'pizza' and 'steak' are kinds of 'food' and, hence, the three terms can be aggregated to show fewer, coarser aspects, even though they are not synonyms. Carenini et al. (2005) used a predefined domain-specific taxonomy to hierarchically aggregate aspect terms, but taxonomies of this kind are often not available, and they are also hard to construct and maintain. By contrast, we use only general-purpose taxonomies (e.g., WordNet), term similarity measures based on general-purpose taxonomies or corpora, and hierarchical clustering.

We define *multi-granular aspect aggregation* to be the task of partitioning a given set of aspect terms (generated by a previous aspect extraction stage) into k non-overlapping clusters, for multiple values of k. A further constraint is that the clusters have to be *consistent* for different k values, meaning that if two aspect terms t_1, t_2 are placed in the same cluster for $k = k_1$, then t_1 and t_2 must also be grouped together (in the same cluster) for every $k = k_2$ with $k_2 < k_1$, i.e., for every coarser grouping. For example, if 'waiter' and 'service' are grouped together for $k = 5$, they must also be grouped together for $k = 4, 3, 2$ and (trivially) $k = 1$, to allow the user to feel that selecting a smaller number of aspect groups (narrowing the height of Fig. 2.1) has the effect of zooming out (without aspect terms jumping unexpectedly to other aspect groups), and similarly for zooming in.[2] This requirement is satisfied by using agglomerative hierarchical clustering algorithms (Manning and Schütze, 1999; Hastie et al., 2001), which in our case produce hierarchies like the ones of Fig. 3.1. By using slices (nodes at a particular depth) of the hierarchies that are closer to the root or the leaves, we obtain fewer

[2] We also require the clusters to be non overlapping to make this zooming in and out metaphor clearer to the user.

Figure 3.1: Aspect hierarchies produced by agglomerative hierarchical clustering.

or more clusters. The vertical dotted lines of Fig. 3.1 illustrate two slices for $k = 4$. By contrast, flat clustering algorithms (e.g., k-means) do not satisfy the consistency constraint for different k values.

Agglomerative clustering algorithms require a measure of the distance between individuals, in our case a measure of how similar two aspect terms are, and a linkage criterion to specify which clusters should be merged to form larger (coarser) clusters. To experiment with different term similarity measures and linkage criteria, we decompose multi-granular aspect aggregation in two processing phases. Phase A fills in a symmetric matrix, like the one of Table 3.1, with scores showing the similarity of each pair of (prominent, i.e., frequent) aspect terms; the matrix in effect defines the distance measure to be used by agglomerative clustering. In Phase B, the aspect terms are grouped into k non-overlapping clusters, for varying values of k, given the matrix of Phase A and a linkage criterion; a hierarchy like the ones of Fig. 3.1 is first formed via agglomerative clustering, and fewer or more clusters (for different values of k) are then obtained by using different slices of the hierarchy, as already discussed. Our two-phase

	food	*fish*	*sushi*	*dishes*	*wine*
food	5	4	4	4	2
fish	4	5	4	2	1
sushi	4	4	5	3	1
dishes	4	2	3	5	2
wine	2	1	1	2	5

Table 3.1: An aspect term similarity matrix.

decomposition can also accommodate non-hierarchical clustering algorithms, provided that the consistency constraint is satisfied, but we consider only agglomerative hierarchical clustering in this thesis.

The decomposition in two phases has three main advantages. Firstly, it allows reusing previous work on term similarity measures (Zhang et al., 2013), which can be used to fill in the matrix of Phase A. Secondly, the decomposition allows different linkage criteria to be experimentally compared (in Phase B) using the same similarity matrix (of Phase A), i.e., the same distance measure. Thirdly, and more importantly, the decomposition leads to high inter-annotator agreement, as we show experimentally. By contrast, in preliminary experiments we found that asking humans to directly evaluate aspect hierarchies produced by hierarchical clustering, or to manually create gold aspect hierarchies led to poor inter-annotator agreement.

We show that existing term similarity measures perform reasonably well in Phase A, especially when combined, but there is a large scope for improvement. We also propose a novel *sense pruning* method for WordNet-based similarity measures, which leads to significant improvements in Phase A. In Phase B, we experiment with agglomerative clustering using four different linkage criteria, concluding that they all perform equally well and that Phase B is almost a solved problem when the gold similarity matrix of Phase A is used; however, further improvements are needed in the similarity measures

of Phase A to produce a sufficiently good similarity matrix. We also make publicly available the datasets of our experiments.

The main contributions of this chapter are: (i) to the best of our knowledge, we are the first to consider multi-granular aspect aggregation (not just merging near-synonyms) in ABSA *without* manually crafted domain-specific ontologies; (ii) we propose a two-phase decomposition that allows previous work on term similarity and hierarchical clustering to be reused and evaluated with high inter-annotator agreement; (iii) we introduce a novel sense pruning mechanism that improves WordNet-based similarity measures; (iv) we provide the first public datasets for multi-granular aspect aggregation; (v) we show that the second phase of our decomposition is almost a solved problem, and that research should focus on the first phase. Although we experiment with customer reviews of products and services, ABSA and the work of this chapter in particular are, at least in principle, also applicable to texts expressing opinions about other kinds of entities (e.g., politicians, organizations).

Section 3.2 below discusses related work. Sections 3.3 and 3.4 present our work for Phase A and B, respectively. Section 3.5 further demonstrates the output of multi-granular aspect aggrefation in three domains. Section 3.6 concludes.

3.2 Related work

Most existing approaches to aspect aggregation aim to produce a single, *flat* partitioning of aspect terms into aspect groups, rather than aspect groups at multiple granularities. The most common approaches (Liu, 2012) are to aggregate only synonyms or near-synonyms, using WordNet (Liu et al., 2005), statistics from corpora (Chen et al., 2006; Bollegala et al., 2007a; Lin and Wu, 2009), or semi-supervised learning (Zhai et al., 2010; Zhai et al., 2011), or to cluster the aspect terms using (latent) topic models (Titov and McDonald, 2008a; Guo et al., 2009; Brody and Elhadad, 2010; Jo and Oh, 2011).

Topic models do not perform better than other methods (Zhai et al., 2010) in aspect aggregation, and their clusters may overlap.[3] The topic model of Titov et al. (2008b) uses two granularity levels; we consider many more (3–10 levels in our experiments).

Carenini et al. (2005) used a *predefined domain-specific* taxonomy and similarity measures to aggregate related terms. Yu et al. (2011b) used a tailored version of an existing taxonomy. By contrast, we assume no predefined domain-specific taxonomy. Kobayashi et al. (2007) proposed methods to extract aspect terms and relations between them, including hierarchical relations. They extract, however, relations by looking for clues in texts (e.g., particular phrases). By contrast, we employ similarity measures and hierarchical clustering, which allows us to group similar aspect terms even when they do not cooccur in texts. Also, in contrast to Kobayashi et al. (2007), we respect the consistency constraint discussed in Section 3.1.

A similar task is taxonomy induction. Cimiano and Staab (2005) automatically construct taxonomies from texts via agglomerative clustering, much as in our Phase B, but not in the context of ABSA, and without trying to learn a similarity matrix first. They also label the hierarchy's concepts, a task we do not consider. Klapaftis and Manandhar (2010) show how word sense induction can be combined with agglomerative clustering to obtain more accurate taxonomies, again not in the context of ABSA. Our sense pruning method was influenced by their work, but is much simpler than their word sense induction. Fountain and Lapata (2012) study unsupervised methods to induce concept taxonomies, without considering ABSA.

[3] Topic models are typically also used to perform aspect term extraction, apart from aspect aggregation, but simple heuristics (e.g., most frequent nouns) often outperform them in aspect term extraction (Liu, 2012; Moghaddam and Ester, 2012), as also discussed in Chapter 2.

	sentences containing n aspect term occurrences			
Domain	$n = 0$	$n \geq 1$	$n \geq 2$	total ($n \geq 0$)
Restaurants	1,099	1,129	829	3,057
Laptops	1,419	823	389	2,631

Table 3.2: Statistics about the aspect term aggregation datasets.

3.3 Phase A

We now discuss our work for Phase A. Recall that in this phase the input is a set of aspect terms and the goal is to fill in a matrix (Table 3.1) with scores showing the similarity of each pair of aspect terms.

3.3.1 Datasets used in Phase A

We used two benchmark datasets that we had previously constructed to evaluate ABSA methods for aspect term extraction (Chapter 2), and aspect score estimation (Chapter 5), but not aspect aggregation. We extended them to support aspect aggregation, and we make them publicly available.[4]

In Table 3.2, we show the number of sentences of our datasets and how many aspect term occurrences they contain. The first column shows that there are many sentences with no aspect terms. The second and third columns show that most sentences contain exactly one aspect term, as already discussed in Chapter 2.

The two original datasets contain manually split sentences from customer reviews of restaurants and laptops, respectively. The datasets are the same as the corresponding ones of Chapter 2 (Section 2.2), but each sentence is also manually annotated as 'subjective' (expressing opinion) or 'objective' (not expressing opinion). In the experiments of this chapter, we use only the 3,057 (out of 3,710) subjective restaurant sentences and

[4]The datasets are available at `http://nlp.cs.aueb.gr/software.html`.

the 2,631 (out of 3,085) subjective laptop sentences.

For each subjective sentence, our datasets show the words that human annotators marked as aspect terms. For example, in "The *dessert* was divine!" the aspect term is 'dessert', and in "Really bad *waiter*." it is 'waiter'. Among the 3,057 subjective restaurant sentences, 1,129 contain exactly one aspect term, 829 more than one, and 1,099 no aspect term; a subjective sentence may express an opinion about the restaurant (or laptop) being reviewed without mentioning a specific aspect (e.g., "Really nice restaurant!"), which is why no aspect terms are present in some subjective sentences. There are 558 distinct multi-word aspect terms and 431 distinct single-word aspect terms in the subjective restaurant sentences. Among the 2,631 subjective sentences of the laptop reviews, 823 contain exactly one aspect term, 389 more than one, and 1,419 no aspect term. There are 273 distinct multi-word aspect terms and 330 distinct single-word aspect terms in the subjective laptop sentences.

From each dataset, we selected the 20 (distinct) aspect terms that the human annotators had annotated most frequently, taking annotation frequency to be an indicator of importance; there are only two multi-word aspect terms ('hard drive', 'battery life') among the 20 most frequent ones in the laptops dataset, and none among the 20 most frequent aspect terms of the restaurants dataset. We then formed all the 190 possible pairs of the 20 terms and constructed an empty similarity matrix (Table 3.1), one for each dataset, which was given to three human judges to fill in (1: strong dissimilarity, 5: strong similarity).[5] For each aspect term, all the subjective sentences mentioning the term were also provided, to help the judges understand how the terms are used in the particular domains (e.g., 'window' and 'Windows' have domain-specific meanings in laptop reviews).

The Pearson correlation coefficient indicated high inter-annotator agreement (0.81

[5]The matrix is symmetric; hence, the judges had to fill in only half of it. The guidelines and an annotation tool that were given to the judges are available upon request.

for restaurants, 0.74 for laptops). We also measured the absolute inter-annotator agreement $a(l_1, l_2)$, defined below, where l_1, l_2 are lists containing the scores (similarity matrix values) of two judges, N is the length of each list, and v_{max}, v_{min} are the largest and smallest possible scores (5 and 1).

$$a(l_1, l_2) = \frac{1}{N} \sum_{i=1}^{N} \left[1 - \frac{|l_1(i) - l_2(i)|}{v_{max} - v_{min}} \right]$$

The absolute interannotator agreement was also high (0.90 for restaurants, 0.91 for laptops).[6] With both measures, we compute the agreement of each judge with the averaged (for each matrix cell) scores of the other two judges, and we report the mean of the three agreement estimates. Finally, we created the *gold* similarity matrix of each dataset by placing in each cell the average scores that the three judges had provided for that cell.

We note that in preliminary experiments, we gave aspect terms to human judges, asking them to group any terms they considered near-synonyms. We then asked the judges to group the aspect terms into fewer, coarser groups by grouping terms that could be viewed as direct hyponyms of the same broader term (e.g., 'pizza' and 'steak' are both kinds of 'food'), or that stood in a hyponym-hypernym relation (e.g., 'pizza' and 'food'). We used the Dice coefficient to measure inter-annotator agreement, and we obtained reasonably good agreement for near-synonyms (0.77 for restaurants, 0.81 for laptops), but poor agreement for the coarser aspects (0.25 and 0.11).[7] In other preliminary experiments, we asked human judges to rank alternative aspect hierarchies that had been produced by applying agglomerative clustering with different linkage criteria to 20 aspect terms, but we obtained very poor inter-annotator agreement (Pearson score -0.83 for restaurants and 0 for laptops). By contrast, the inter-annotator agreement on the similarity matrices was reasonably high, as already discussed.

[6] The Pearson correlation ranges from -1 to 1, whereas the absolute inter-annotator agreement ranges from 0 to 1.

[7] The Dice coefficient ranges from 0 to 1. There was a very large number of possible responses the judges could provide and, hence, it would be inappropriate to use Cohen's K.

3.3.2 Phase A methods

We employed five term similarity measures. The first two are WordNet-based (Fellbaum, 1998; Budanitsky and Hirst, 2006).[8] The next two combine WordNet with statistics from corpora. The fifth one is a corpus-based distributional similarity measure.

The first measure is *Wu and Palmer's* (1994). It is actually a sense similarity measure (a term may have multiple senses). Given two senses $s_{ij}, s_{i'j'}$ of terms $t_i, t_{i'}$, the measure is defined as follows:

$$WP(s_{ij}, s_{i'j'}) = 2 \cdot \frac{depth(lcs(s_{ij}, s_{i'j'}))}{depth(s_{ij}) + depth(s_{ij})},$$

where $lcs(s_{ij}, s_{i'j'})$ is the *least common subsumer*, i.e., the most specific common ancestor of the two senses in WordNet, and $depth(s)$ is the depth of sense (synset) s in WordNet's hierarchy.

Most terms have multiple senses, however, and word sense disambiguation methods (Navigli, 2009) are not yet robust enough. Hence, when given two aspect terms $t_i, t_{i'}$, rather than particular senses of the terms, a simplistic *greedy* approach is to compute the similarities of all the possible pairs of senses $s_{ij}, s_{i'j'}$ of $t_i, t_{i'}$, and take the similarity of $t_i, t_{i'}$ to be the maximum similarity of the sense pairs (Bollegala et al., 2007b; Zesch and Gurevych, 2010). We use this greedy approach with all the WordNet-based measures, but we also propose a sense pruning mechanism below, which improves their performance. In all the WordNet-based measures, if a term is not in WordNet, we take its similarity to any other term to be zero.[9]

The second measure, $PATH(s_{ij}, s_{i'j'})$, is simply the inverse of the length (plus one) of the shortest path connecting the senses $s_{ij}, s_{i'j'}$ in WordNet (Zhang et al., 2013). Again, the greedy approach can be used with terms having multiple senses.

[8] See `http://wordnet.princeton.edu/`.

[9] This never happened in the restaurants dataset. In the laptops dataset, it only happened for 'hard drive' and 'battery life'. We use the NLTK implementation of the first four measures (see `http://nltk.org/`) and our own implementation of the distributional similarity measure.

The third measure is *Lin's* (1998), defined as:

$$LIN(s_{ij}, s_{i'j'}) = \frac{2 \cdot ic(lcs(s_{ij}, s_{i'j'}))}{ic(s_{ij}) + ic(s_{i'j'})},$$

where $s_{ij}, s_{i'j'}$ are senses of terms $t_i, t_{i'}$, $lcs(s_{ij}, s_{i'j'})$ is the least common subsumer of $s_{ij}, s_{i'j'}$ in WordNet, and $ic(s) = -\log P(s)$ is the *information content* of sense s (Pedersen et al., 2004), estimated from a corpus. When the corpus is not sense-tagged, we follow the common approach of treating each occurrence of a word as an occurrence of all of its senses, when estimating $ic(s)$.[10] We experimented with two variants of Lin's measure, one where the $ic(s)$ scores were estimated from the Brown corpus (Marcus et al., 1993), and one where they were estimated from the (restaurant or laptop) reviews of our datasets.

The fourth measure is *Jiang and Conrath's* (1997), defined below. Again, we experimented with two variants of $ic(s)$, as above.

$$JCN(s_{ij}, s_{i'j'}) = \frac{1}{ic(s_{ij}) + ic(s_{i'j'}) - 2 \cdot lcs(s_{ij}, s_{i'j'})}$$

For all the above WordNet-based measures, we experimented with a *sense pruning* mechanism, which discards some of the senses of the aspect terms, before applying the greedy approach. For each aspect term t_i, we consider all of its WordNet senses s_{ij}. For each s_{ij} and each other aspect term $t_{i'}$, we compute (using *PATH*) the similarity between s_{ij} and each sense $s_{i'j'}$ of $t_{i'}$, and we consider the *relevance* of s_{ij} to $t_{i'}$ to be:[11]

$$rel(s_{ij}, t_{i'}) = \max_{s_{i'j'} \in senses(t_{i'})} PATH(s_{ij}, s_{i'j'})$$

The relevance of s_{ij} to *all* of the N other aspect terms $t_{i'}$ is taken to be:

$$rel(s_{ij}) = \frac{1}{N} \cdot \sum_{i' \neq i} rel(s_{ij}, t_{i'})$$

[10] See `http://www.d.umn.edu/~tpederse/Data/README-WN-IC-30.txt`. We use the default counting.

[11] We also experimented with other similarity measures when computing $rel(s_{ij}, t_{i'})$, instead of *PATH*, but there was no significant difference. We use NLTK to tokenize, remove punctuation, and stop-words.

For each aspect term t_i, we retain only its senses s_{ij} with the top $rel(s_{ij})$ scores, which tends to prune senses that are very irrelevant to the particular domain (e.g., laptops). This sense pruning mechanism is novel, and we show experimentally that it improves the performance of all the WordNet-based similarity measures we examined.

We also implemented a *distributional similarity* measure (Harris, 1968; Padó and Lapata, 2007; Cimiano et al., 2009; Zhang et al., 2013). Following Lin and Wu (2009), for each aspect term t, we create a vector $\vec{v}(t) = \langle \text{PMI}(t, w_1), \ldots, \text{PMI}(t, w_n) \rangle$. The vector components are the Pointwise Mutual Information scores of t and each word w_i of a corpus:

$$\text{PMI}(t, w_i) = -\log \frac{P(t, w_i)}{P(t) \cdot P(w_i)}$$

We treat $P(t, w_i)$ as the probability of t, w_i cooccurring in the same sentence, and we use the (laptop or restaurant) reviews of our datasets as the corpus to estimate the probabilities. The distributional similarity $DS(t, t')$ of two aspect terms t, t' is the cosine similarity of $\vec{v}(t), \vec{v}(t')$.[12]

Finally, we tried combinations of the similarity measures: *AVG* is the average of all five; *WN* is the average of the first four, which employ WordNet; and *WNDS* is the average of *WN* and *DS*.[13].

3.3.3 Phase A experimental results

Each similarity measure was evaluated by computing its Pearson correlation with the scores of the gold similarity matrix. Table 3.3 shows the results.

Our sense pruning consistently improves all four WordNet-based measures. It does not apply to *DS*, which is why the *DS* results are identical with and without pruning. A

[12]We also experimented with Euclidean distance, a normalized PMI (Bouma, 2009), and the Brown corpus, but there was no improvement.

[13]The range of all the similarity measures is in $[0, 1]$. We also experimented with regression, but there was no improvement. We experimented both with linear regression and Support Vector Regression (SVR) (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000), using leave one out cross validation.

	without SP		**with SP**	
Method	*Restaurants*	*Laptops*	*Restaurants*	*Laptops*
WP	0.475	0.216	0.502	0.265
PATH	**0.524**	0.301	0.529	0.332
LIN@domain	0.390	0.256	0.456	0.343
LIN@Brown	0.434	0.329	0.471	0.391
JCN@domain	0.467	0.348	0.509	0.448
JCN@Brown	0.403	0.469	0.419	0.539
DS	0.283	**0.517**	(0.283)	(0.517)
AVG	0.499	0.352	0.537	0.426
WN	0.490	0.328	0.530	0.395
WNDS	0.523	0.453	**0.545**	**0.546**

Table 3.3: Phase A results (Pearson correlation to gold similarities) *with* and *without* our sense pruning (SP).

paired t test indicates that the other differences (with and without pruning) of Table 3.3 are statistically significant ($p < 0.05$). We used the senses with the top five $rel(s_{ij})$ scores for each aspect term t_i during sense pruning. We also experimented with keeping fewer senses, but the results were inferior or there was no improvement.

Lin's measure performed better when information content was estimated on the (much larger, but domain-independent) Brown corpus (*LIN@Brown*), as opposed to using the (domain-specific) reviews of our datasets (*LIN@domain*), but we observed no similar consistent pattern for *JCN*. Given its simplicity, *PATH* performed remarkably well in the restaurants dataset; it was the best measure (including combinations) without sense pruning, and the best uncombined measure with sense pruning. It performed worse, however, compared to several other measures in the laptops dataset. Similar comments apply to *WP*, which is among the top-performing uncombined measures in restaurants, both with and without sense pruning, but the worst overall measure in laptops. *DS* is the best overall measure in laptops when compared to measures without sense pruning, and the third best overall when compared to measures that use sense pruning, but the worst overall in restaurants both with and without pruning. *LIN* and *JCN*, which use both WordNet and corpus statistics, have a more balanced performance across the two datasets, but they are not top-performers in any of the two. *Combinations of similarity measures seem more stable across domains*, as the results of *AVG*, *WN*, and *WNDS* indicate, though experiments with more domains are needed to investigate this issue. *WNDS is the best overall method with sense pruning*, and among the best three methods without pruning in both datasets.

To get a better view of the performance of *WNDS* with sense pruning, i.e., the best overall measure of Table 3.3, we compared it to two state of the art semantic similarity systems. First, we applied the system of Han et al. (2013), one of the best systems of the recent *Sem 2013 semantic textual similarity competition (Agirre et al., 2013), to our Phase A data. The performance (Pearson correlation with gold similarities) of

Method	Restaurants	Laptops
Han et al. (2013)	0.450	0.471
Word2Vec	0.434	0.485
WNDS with SP	**0.545**	**0.546**
Judge 1	0.913	0.875
Judge 2	0.914	0.894
Judge 3	0.888	0.924

Table 3.4: Phase A results (Pearson correlation to gold similarities) of *WNDS* with SP against state of the art semantic similarity systems and human judges.

the same system on the widely used *WordSim353* word similarity dataset (Agirre et al., 2009) is 0.73, much higher than the same system's performance on our Phase A data (see Table 3.4), which suggests that our data are more difficult.[14]

We also employed the recent *Word2Vec* system, which computes continuous vector space representations of words from large corpora and has been reported to improve results in word similarity tasks (Mikolov et al., 2013a; Mikolov et al., 2013b; Mikolov et al., 2013c). We used the English Wikipedia to compute word vectors with 200 features.[15] The similarity between two aspect terms was taken to be the cosine similarity of their vectors. This system performed better than Han et al.'s with laptops, but not with restaurants.

Table 3.4 shows that *WNDS (with sense pruning) performed clearly better than the system of Han et al. and Word2Vec*. Table 3.4 also shows the Pearson correlation of

[14]The system of Han et al. (2013) is available from `http://semanticwebarchive.cs.umbc.edu/SimService/`; we use the STS similarity.

[15]*Word2Vec* is available from `https://code.google.com/p/word2vec/`. We used the continuous bag of words model with default parameters, the first billion characters of the English Wikipedia, and the preprocessing of `http://mattmahoney.net/dc/textdata.html`.

each judge's scores to the gold similarity scores, as an indication of the best achievable results. Although *WNDS* (with sense pruning) performs reasonably well in both domains (recall that the Pearson correlation ranges from -1 to 1), *there is large scope for improvement.*

3.4 Phase B

In Phase B, the aspect terms are to be grouped into k non-overlapping clusters, for varying values of k, given a Phase A similarity matrix. We experimented with both the gold similarity matrix of Phase A and similarity matrices produced by *WNDS* with sense pruning, the best Phase A method.

3.4.1 Phase B methods

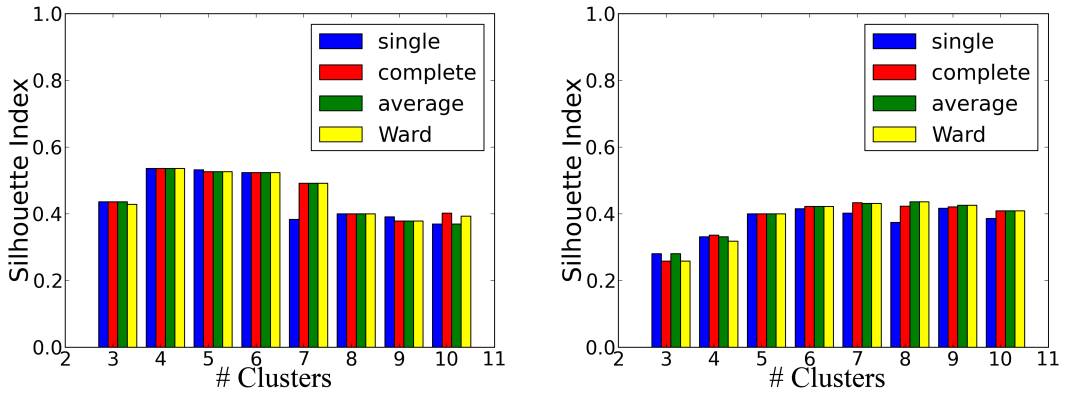
We experimented with agglomerative clustering and four linkage criteria: *single*, *complete*, *average*, and *Ward* (Manning and Schütze, 1999; Hastie et al., 2001). Let $d(t_1,t_2)$ be the distance of two individual instances t_1,t_2; in our case, the instances are aspect terms and $d(t_1,t_2)$ is the inverse of the similarity of t_1,t_2, defined by the Phase A similarity matrix (gold or produced by *WNDS*). Different linkage criteria define differently the distance of two clusters $D(C_1,C_2)$, which affects the choice of clusters that are merged to produce coarser (higher-level) clusters:

$$\begin{aligned} D_{single}(C_1,C_2) &= \min_{t_1\in C_1,\ t_2\in C_2} d(t_1,t_2) \\ D_{compl}(C_1,C_2) &= \max_{t_1\in C_1,\ t_2\in C_2} d(t_1,t_2) \\ D_{avg}(C_1,C_2) &= \frac{1}{|C_1||C_2|}\sum_{t_1\in C_1}\sum_{t_2\in C_2} d(t_1,t_2) \end{aligned}$$

Complete linkage tends to produce more compact clusters, compared to single linkage, with average linkage being in between. Ward minimizes the total in-cluster variance;

consult Milligan (1980) for further details.[16]

3.4.2 Phase B experimental results



(a) restaurants (b) laptops

Figure 3.2: Silhouette Index (SI) results for Phase B, using the **gold** similarity matrix of Phase A.

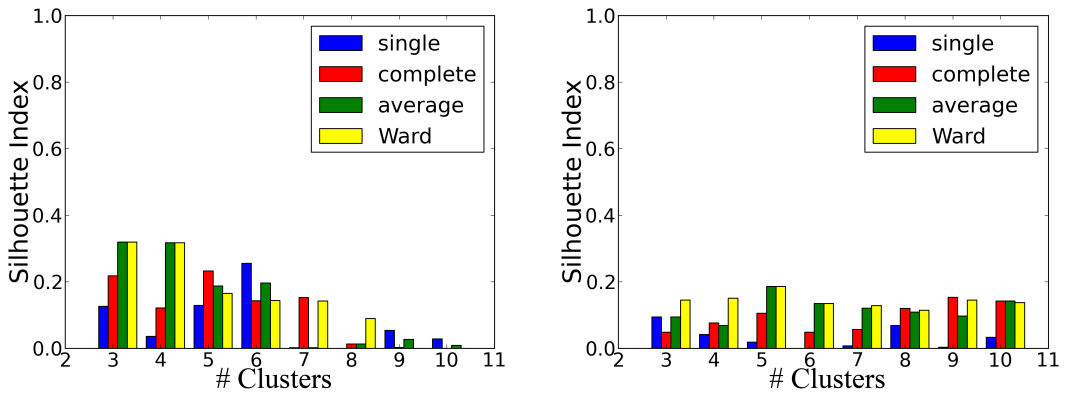
To evaluate the k clusters produced at each aspect granularity by the different linkage criteria, we used the *Silhouette Index* (*SI*) (Rousseeuw, 1987), a cluster evaluation measure that considers both inter- and intra-cluster coherence.[17] Given a set of clusters $\{C_1, \ldots, C_k\}$, each $SI(C_i)$ is defined as:

$$SI(C_i) = \frac{1}{|C_i|} \cdot \sum_{j=1}^{|C_i|} \frac{b_j - a_j}{\max(b_j, a_j)},$$

where a_j is the mean distance from the j-th instance of C_i to the other instances in C_i, and b_j is the mean distance from the j-th instance of C_i to the instances in the cluster

[16] We used the SciPy implementations of agglomerative clustering with the four criteria (see `http://www.scipy.org`), relying on *maxclust* to obtain the slice of the resulting hierarchy that leads to k (or approx. k) clusters.

[17] We used the *SI* implementation of Pedregosa et al. (2011); see `http://scikit-learn.org/`. We also experimented with the Dunn Index (Dunn, 1974) and the Davies-Bouldin Index (1979), but we obtained similar results.



(a) restaurants (b) laptops

Figure 3.3: Silhouette Index (SI) results for Phase B, using the **WNDS (with SP)** similarity matrix of Phase A.

nearest to C_i. Then:

$$SI(\{C_1,\ldots,C_k\}) = \frac{1}{k}\cdot\sum_{i=1}^{k} SI(C_i)$$

We always use the correct (gold) distances of the instances (terms) when computing the *SI* scores, but the clusters being evaluated may not have been produced using the gold distances (e.g., when using the similarity matrix produced by a Phase A method).

As shown in Fig. 3.2, *no linkage criterion clearly outperforms the others, when the gold matrix of Phase A is used to produce the clusters*; all four criteria perform reasonably well. Note that the *SI* ranges from -1 to 1, with higher values indicating better clustering. Figure 3.3 shows that *when the similarity matrix of WNDS (with SP) is used to produce the clusters, the SI scores deteriorate significantly*; again, there is no clear winner among the linkage criteria, but average and Ward seem to be overall better than the others.

In a final experiment, we showed clusterings of varying granularities (k values) to four human judges (graduate CS students). The clusterings were produced by two systems: one that used the *gold similarity matrix* of Phase A and agglomerative clustering

with average linkage in Phase B, and one that used the *similarity matrix of WNDS* (with SP) and again agglomerative clustering with average linkage. We showed all the clusterings to all the judges. Each judge was asked to evaluate each clustering on a 1–5 scale (1: totally unacceptable, 5: perfect). We measured the absolute inter-annotator agreement, as in Section 3.3.1, and found high agreement in all cases (0.93 and 0.83 for the two systems, respectively, in restaurants; 0.85 for both systems in laptops).[18]

Figure 3.4 shows the average human scores of the two systems for different granularities. The judges considered the aspect groups (clusters) always perfect or near-perfect when the gold similarity matrix of Phase A was used, but they found the aspect groups to be of rather poor quality when the similarity matrix of the best Phase A measure was used. These results, along with those of Fig. 3.2–3.3, show that *more effort needs to be devoted in future work to improving the similarity measures of Phase A, whereas Phase B is in effect an almost solved problem*, if a good similarity matrix is available.

3.5 Demonstration

In this section, we further demonstrate the use of multi-granular aspect aggregation. We first use the similarity matrix which was generated in Phase A by the best performing method (*WNDS* with SP), along with agglomerative average linkage clustering, to produce aspect term hierarchies for domains (laptops, restaurants, hotels).[19] In Fig. 3.5 we show the generated aspect term hierarchies. Below, we show the clusterings produced by dissecting the aspect term hierarchies, when 3, 5, and 5 clusters (e.g., rows in a mobile phone) are requested.

[18]The Pearson correlation cannot be computed, as several judges gave the same rating to the first system, for all k.

[19]The hotels dataset for aspect aggregation is based on the corresponding ATE htoels dataset of Chapter 2. It is still under validation and is being further annotated for the ABSA SEMEVAL 2015 task. We use the current, preliminary form of the dataset in this section.

Restaurants (*WNDS* with SP in Phase A, average linkage clustering in Phase B):,

- for 3 clusters: (prices, price), (dishes, food, menu, sushi, lunch, dinner, portions, table, pizza, meal, drinks), (decor, bar, service, atmosphere, place, staff, ambience)
- for 5 clusters: (prices, price), (dishes, food, menu, sushi, lunch, dinner, portions, table, pizza, meal, drinks), (atmosphere, place, ambience), (bar, service, staff), (decor)
- for 7 clusters: (prices, price), (portions, drinks), (sushi, pizza), (dishes, food, menu, lunch, dinner, table, meal), (atmosphere, place, ambience), (bar, service, staff), decor)

Laptops (*WNDS* with SP in Phase A, average linkage clustering in Phase B),

- for 3 clusters: (warranty, battery, screen, keyboard, graphics, battery life, hard drive, memory), (quality, use, price, speed, size), (runs, features, programs, windows, performance, software, applications)
- for 5 clusters: (battery, screen, keyboard, battery life, hard drive, memory), (graphics), (warranty), (quality, use, price, speed, size), (runs, features, programs, windows, performance, software, applications)
- for 7 clusters: (battery, screen, keyboard, battery life, hard drive, memory), (graphics), (warranty), (quality, use, price), (speed, size), (programs, windows, software, applications), (runs, features, performance)

Hotels (*WNDS* with SP in Phase A, average linkage clustering in Phase B),

- for 3 clusters: (noise, price, views, view), (shower, bed, beds), (bathroom, bar, room, service, restaurant, food, wifi, rooms, location, internet, breakfast, staff, pool)

- for 5 clusters: (noise, price, views, view), (shower, bed, beds), (wifi, internet), (food, breakfast), (bathroom, bar, room, service, restaurant, rooms, location, staff, pool)

- for 7 clusters: (views, view), (noise, price), (shower, bed, beds), (wifi, internet), (food, breakfast), (bathroom, room, rooms), (bar, service, restaurant, location, staff, pool)

Next, we show the clusterings produced by dissecting the aspect term hierarchies, which are generated when the gold similarity matrix is used in Phase A and, again, average linkage clustering in Phase B (Fig. 3.1). No clusterings are shown for the hotels domain, because gold annotations have not been constructed yet.
Restaurants (gold similarity matrix in Phase A, average linkage clustering in Phase B):,

- for 3 clusters: (portions, dishes, food, menu, sushi, meal, pizza, drinks, fish, wine), (price, prices), (staff, decor, service, waiter, atmosphere, place, ambience, table)

- for 5 clusters: (drinks, wine), (portions, dishes, food, menu, sushi, meal, pizza, fish), (price, prices), (staff, service, waiter), (decor, atmosphere, place, ambience, table)

- for 7 clusters: (drinks, wine), (portions, meal), (dishes, food, menu, sushi, pizza, fish), (price, prices), (staff, service, waiter), (decor, atmosphere, place, ambience), (table)

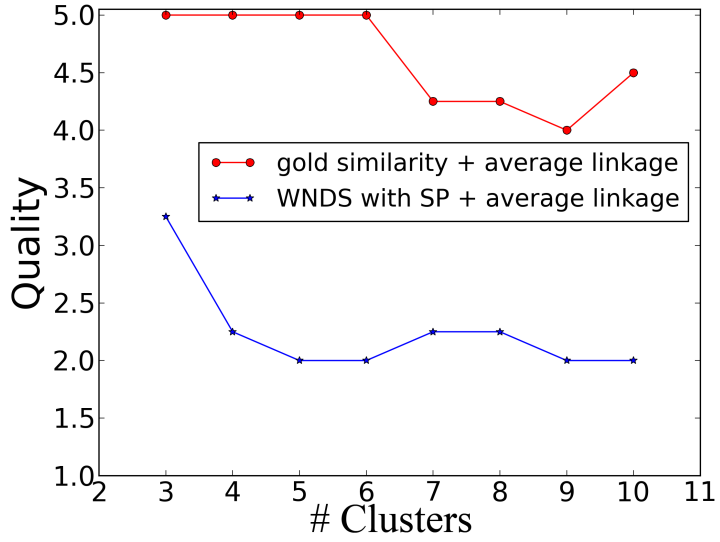
Laptops (gold similarity matrix in Phase A, average linkage clustering in Phase B),

- for 3 clusters: (programs, windows, applications, software), (warranty, service, price), (graphics, features, battery, quality, screen, keyboard, battery life, design, hard drive, memory, performance, speed, size)

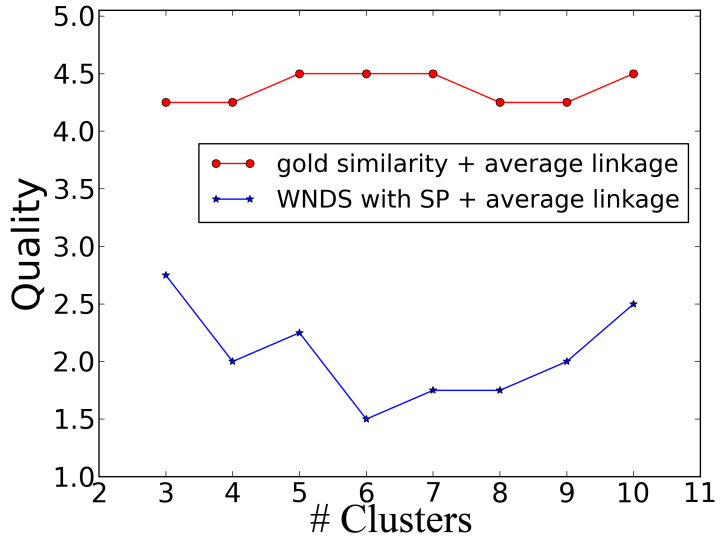
- for 5 clusters: (programs, windows, applications, software), (warranty, service, price), (battery, battery life), (quality, performance, speed), (graphics, features, screen, keyboard, design, hard drive, memory, size)
- for 7 clusters: (programs, windows, applications, software), (warranty, service, price), (battery, battery life), (quality, performance, speed), (design, size), (hard drive, memory), (graphics, features, screen, keyboard)

3.6 Conclusions

We considered a new, more demanding form of aspect aggregation in ABSA, which aims to aggregate aspect terms at multiple granularities, as opposed to simply merging near-synonyms, and without assuming that manually crafted domain-specific ontologies are available. We decomposed the problem in two processing phases, which allow previous work on term similarity and hierarchical clustering to be reused and evaluated appropriately with high inter-annotator agreement. We showed that the second phase, where we used agglomerative clustering, is an almost solved problem, whereas further research is needed in the first phrase, where term similarity measures are employed. We also introduced a sense pruning mechanism that significantly improves WordNet-based similarity measures, leading to a measure that outperforms state of the art similarity methods in the first phase of our decomposition. We also made publicly available the datasets of our experiments.

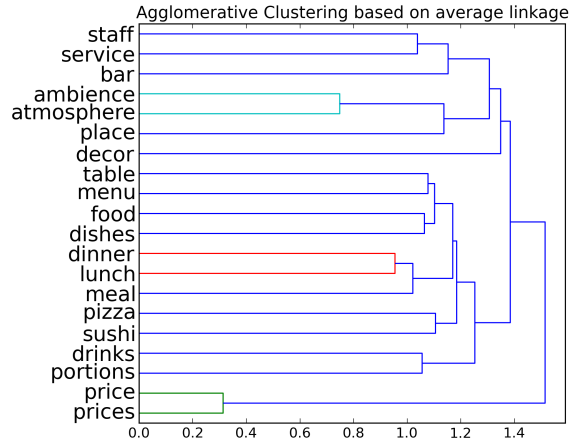


(a) Restaurants

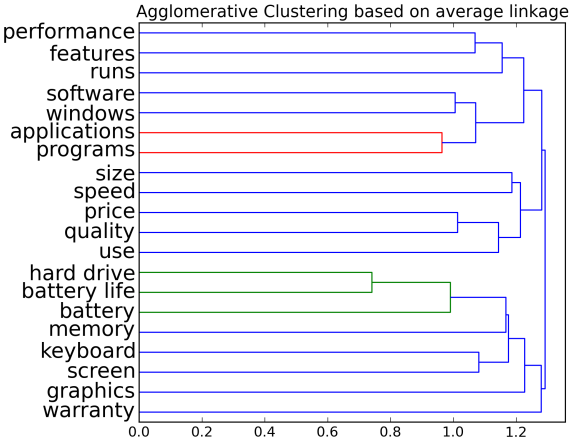


(b) Laptops

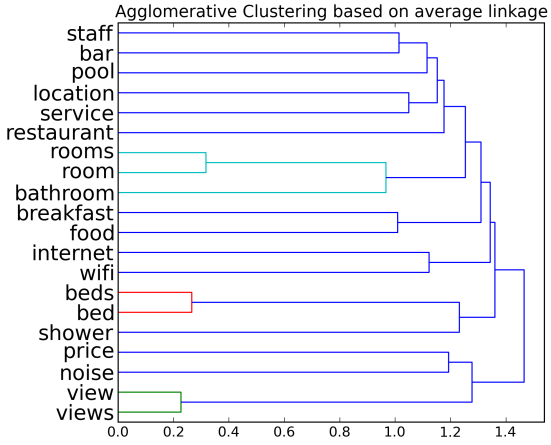
Figure 3.4: Human evaluation of aspect groups (clusters of aspect terms) at different granularities (number of clusters).



(a) Restaurants

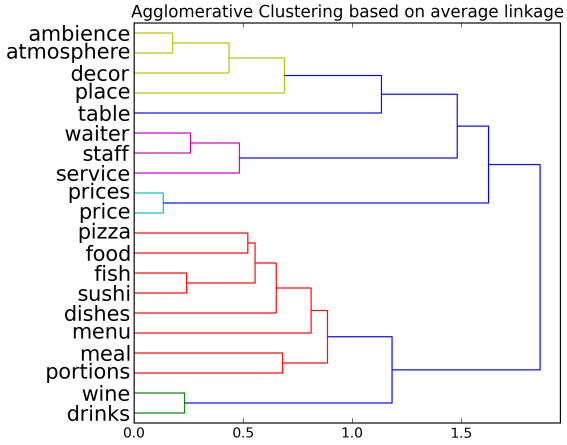


(b) Laptops

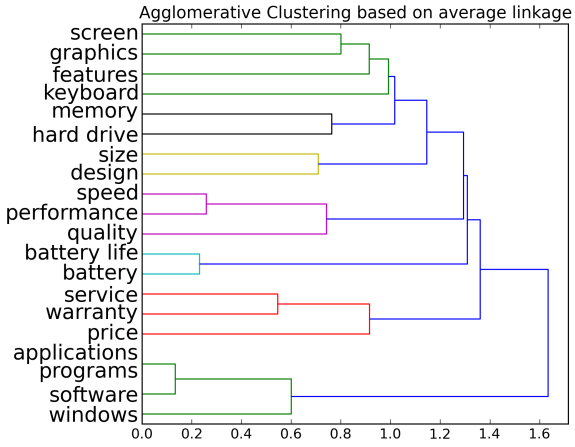


(c) Hotels

Figure 3.5: Aspect term hierarchies for our three domain datasets, generated with *WNDS* with SP in Phase A and agglomerative average linkage clustering in Phase B.



(a) Restaurants



(b) Laptops

Figure 3.6: Aspect term hierarchies for two of our domain datasets, generated with gold similarity matrix in Phase A and agglomerative average linkage clustering in Phase B.

Chapter 4

Message-level Sentiment Estimation[1]

4.1 Introduction

Classifying texts by sentiment polarity and possibly also by sentiment intensity (e.g., strongly negative, mildly positive) is a popular research topic (Liu, 2012; Pang and Lee, 2005; Tsytsarau and Palpanas, 2012). A large part of previous work on sentiment classification is concerned with assigning sentiment labels to entire sentences or, more generally, messages (especially tweets) and several benchmark datasets and competitions exist for this task. In the context of aspect-based sentiment analysis (ABSA), we need to determine the sentiment polarity (and possibly intensity) for each mentioned aspect term (or coarser aspect) of a target entity. In this chapter, however, we take a digression to present first a message-level sentiment classification system that was developed during the work of this thesis. In the next chapter, we show how this system and, more generally, the experience gained by developing the message-level classifier can also be applied to ABSA to estimate the sentiment polarities of opinions about particular aspect terms.[2]

[1] Summaries of this chapter have been published (Malakasiotis et al., 2013; Karampatsis et al., 2014).

[2] The team that developed the system of this chapter comprises the author, Ion Androutsopoulos, Rafael Michael Karampatsis, and Makis Malakasiotis. In 2013, Nantia Makrynioti was also part of the

The system of this chapter operates in two stages and classifies short texts (e.g., sentences, SMS, tweets), hereafter called 'messages', as positive, negative, or neutral. In the first stage, the system classifies each message as 'subjective' (i.e., carrying positive overall sentiment, negative sentiment, or both) or neutral (i.e., carrying no sentiment, e.g., "it has a 42-inch screen"); messages of the latter type are also called 'objective'. In the second stage, subjective messages are further classified as positive or negative. This decomposition has two main advantages. Firstly, subjectivity detection (separating subjective from objective messages) can be useful on its own; for example, it may be desirable to highlight sentences of reviews (or other texts) that express subjective opinions. Secondly, the two-stage decomposition allowed us to address the class imbalance of the datasets we experimented with, where there are more neutral messages than positive and negative. Our system participated in the subtask 'Message Polarity Classification' (Wilson et al., 2005) of the task of 'Sentiment Analysis in Twitter' in SEMEVAL 20013 and 2014.[3] The organisers provided test sets containing tweets, but also test sets of a different nature (e.g., SMS messages). Our system was ranked high in all the test sets, in both years, which shows that it is able to generalize well.

4.2 Message-level sentiment estimation datasets

Before we proceed with our system description, we briefly discuss the data released by the SEMEVAL organisers, for the subtask we participated in.

In SEMEVAL 2013, the training set consisted of a set of tweet IDs (IDs that can in principle be used to download the corresponding tweets), along with the correct sentiment labels of the tweets (positive, negative, neutral). In order to address copyright concerns, rather than releasing the actual tweets, the organisers provided a Python script

team.

[3]Consult `http://alt.qcri.org/semeval20014/task9/` and `http://www.cs.york.ac.uk/semeval-2013/task2/` for further details.

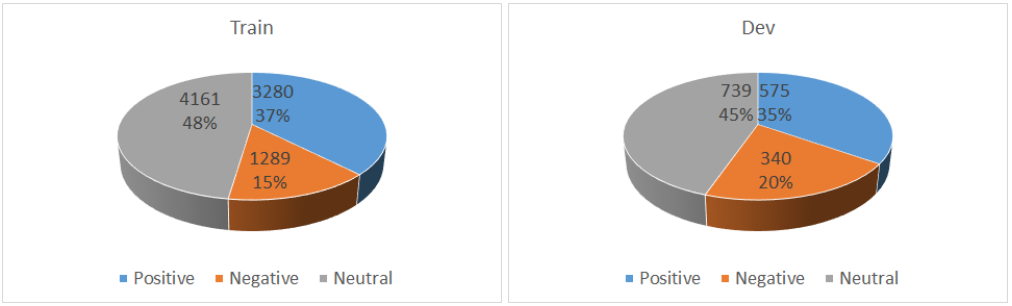


Figure 4.1: Train and development data class distribution in SEMEVAL 2013.

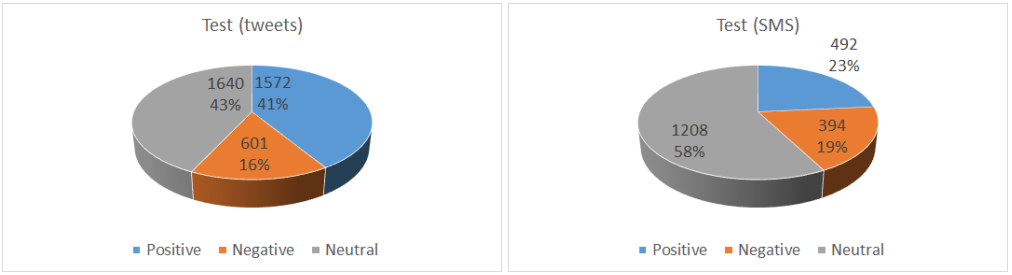


Figure 4.2: Test data class distribution in SEMEVAL 2013.

that each participant could use to download the tweets via their IDs. This led to a situation where different participants had slightly different versions of the training set, since tweets may often become unavailable due to a number of reasons. For the test set, the organisers downloaded and provided the tweets directly. A separate test set with SMS messages was also provided by the organisers to measure the performance of the systems on messages of a different type than the type of messages they had been trained on. No SMS training and development data were provided. A first analysis of the training data and development data (tweets) revealed a class imbalance problem Figures 4.1 and 4.2. Specifically, the training data we downloaded contained 8730 tweets (3280 positive, 1289 negative, 4161 neutral), while the development set contained 1654 tweets (575 positive, 340 negative, 739 neutral). The 2013 test data were released after the end of the competition. The tweets test dataset contained 3813 tweets (1572 posi-

tive, 601 negative, 1640 neutral). The SMS test dataset contained 2094 messages (492 positive, 394 negative, 1208 neutral).

In the second year (SEMEVAL 2014), the training and development data of 2013 were used for training, i.e., the development set of the previous year was added to the training set. The test data of 2013 (tweets and SMS) were used for development in 2014. The class imbalance problem remained. The test data of 2014 contained 8987 messages (tagged as positive, negative, or neutral), comprising the following:

- LJ_{14}: 2000 sentences from LIVEJOURNAL.
- SMS_{13}: SMS test data from 2013.
- TW_{13}: Twitter test data from 2013.
- TW_{14}: 2000 new tweets.
- $TWSARC_{14}$: 100 tweets containing sarcasm.

Again, the details of the 2014 test data were made available to the participants only after the end of the competition. Strangely, SMS_{13} and TW_{13} were used both as development and test data in 2014.

4.3 Our two-stage system

Our system follows a two-stage approach. During the first stage, we detect whether a message expresses some sentiment ('subjective' message) or not; this process is often called subjectivity detection. In the second stage, we classify the 'subjective' messages of the first stage as 'positive' or 'negative' (Figure 4.3). Both stages utilize a Support Vector Machine (SVM) (Vapnik, 1998) classifier with a linear kernel.[4] Similar approaches have also been proposed in previous work (Pang and Lee, 2004; Wilson et

[4]We used the LIBLINEAR implementation (Fan et al., 2008).

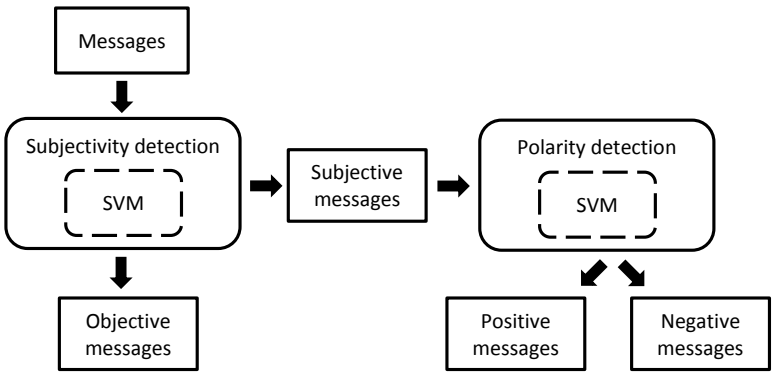


Figure 4.3: Our two–stage message-level sentiment classifier.

al., 2005; Barbosa and Feng, 2010; Malakasiotis et al., 2013). We note that the two-stage approach alleviates the class imbalance problem (Figures 4.1 and 4.2), since each one of the two classifiers is trained on a more balanced dataset (of two classes).

4.3.1 Data preprocessing

A very essential part of our system is data preprocessing. At first, each message is passed through a twitter-specific tokenizer and part-of-speech (POS) tagger (Owoputi et al., 2013) to obtain the tokens and the corresponding POS tags, which are necessary for some of the features that we use (discussed below). We then use a slang dictionary to replace any slang expression with the corresponding non-slang expression.[5] We also normalize the text of each message by replacing every token (excluding punctuation etc.) that is not present in a general purpose large English dictionary with the most similar word of the dictionary.[6] Similarity is measured as edit distance and we use a trie data structure and dynamic programming to efficiently compute the distance of each token of the message to the words of the dictionary (Karampatsis, 2012).

[5]See `http://www.noslang.com/dictionary/`.

[6]We used the OPENOFFICE dictionary, available from `https://www.openoffice.org/lingucomponent/download_dictionary.html`.

4.3.2 Sentiment lexica

Another key characteristic of our system is the use of sentiment lexica. We used the following lexica:

– HL (Hu and Liu, 2004), which is a list of approximately 2006 positive and 4783 negative opinion words for English.

– The SENTIWORDNET lexicon with POS tags (Baccianella et al., 2010), which assigns to each synset (sense) of WordNet three sentiment scores, positivity, negativity, objectivity.[7] For each word, we take the sentiment score of its most frequent sense (listed first in WordNet) for the given POS tag.

– A version of SENTIWORDNET (Baccianella et al., 2010) without POS tags (which we have constructed), where, for each word, we average the sentiment scores of its POS tags.

– AFINN (Nielsen, 2011), which is a list of 2477 English words and phrases, rated for valence with an integer between minus five (negative) and plus five (positive).

– The MPQA subjectivity lexicon (Wilson et al., 2005), which is a list of 8222 POS-tagged words, annotated towards their polarity (positive, negative, or neutral) and their intensity of subjectivity (weak or strong).

– NRC Emotion lexicon (Mohammad and Turney, 2013), which contains information about 14,177 word types, including whether the word is positive or negative and whether it has associations with eight basic emotions (joy, sadness, anger,

[7] In WordNet, which is a large lexical database of English, nouns, verbs, adjectives, and adverbs are grouped into sets of synonyms (synsets). Each set contains words that can be used with the same sense. Words tgat have multipe senses appear in multiple sets. Hence, the sets (synsets) can be thought of as representing word senses. Synsets are interlinked by means of conceptual-semantic and lexical relations. See `http://wordnet.princeton.edu/`.

fear, surprise, anticipation, trust, disgust). For each word, we compute three scores; the number of positive emotions (joy, surprise, anticipation, trust) associated with the word, plus one if the word is positive; the number of negative emotions (sadness, fear, anger, disgust), plus one if the word is negative; a subjectivity score, which is the sum of our positive and negative scores. We use only the 6,464 word types which have at least one non zero score.

– The NRC Hashtag lexicon (Mohammad et al., 2013), which is a list of 54,129 unigrams (i.e., words), 316,531 bigrams (i.e., unigram–unigram terms), and 480,000 pairs of unigrams and bigrams (i.e., bigrams, unigram–bigram terms, bigram–unigram terms, or bigram–bigram pairs), annotated with sentiment scores. The sentiment score is a real number, which is the association of the term with positive sentiment minus the association of the term with negative sentiment.[8]

– The NRC S140 lexicon (Mohammad et al., 2013), which is a list of words with associations to positive and negative sentiments. It has the same format as the NRC Hashtag lexicon, but it was created from 1,6 million tweets, and only emoticons were used as positive and negative seed words.

– The three lexica created from the SEMEVAL 2013 training data by Malakasiorits et al. (2013). These lexica were constructed as follows. For each sentiment class (positive, negative, neutral), we obtained the 100 most important words from the training set (based on Chi Squared feature selection). Few terms were manually removed, to yield a lexicon with the 94 most important tokens appearing in positive tweets, a lexicon with the 96 most important tokens appearing in negative tweets, and a lexicon with the 94 most important tokens appearing in neutral

[8]The association of a term with the positive (or negative) sentiment was measured from 775K tweets, as the pointwise mutual information score of the term with a list of 32 positive (or 36 negative) hashtagged seed words (e.g., #good or #bad).

tweets.

In the case of the MPQA Lexicon, we applied preprocessing to obtain various sub–lexica. The MPQA Lexicon contains words that often indicate subjectivity. A word that in most contexts expresses sentiment is considered to be 'strong' subjective, otherwise it is considered 'weak' subjective (i.e., it has specific subjective usages). We first split the MPQA lexicon in two smaller ones, one containing strong and one containing weak subjective words. Moreover, Wilson (2005) also reports the polarity of each MPQA word out of context (prior polarity), which can be positive, negative or neutral. Hence, we further split the two (sub)lexica into four smaller ones, also taking into account the prior polarity of each expression, obtaining the following eight MPQA-based (sub)lexica:

S_+ : Contains strong subjective words with positive prior polarity (e.g., 'charismatic').

S_- : Contains strong subjective words with negative prior polarity (e.g., 'abase').

$S_\pm$: Contains strong subjective words with either positive or negative prior polarity (e.g., 'abase' or 'charismatic').

S_0 : Contains strong subjective words with neutral prior polarity (e.g., 'disposition').

W_+ : Contains weak subjective words with positive prior polarity (e.g., 'drive').

W_- : Contains weak subjective words with negative prior polarity (e.g., 'dusty').

$W_\pm$: Contains weak subjective words with either positive or negative prior polarity (e.g., 'drive' or 'dusty').

W_0 : Contains weak subjective expressions with neutral prior polarity (e.g., 'duty').

4.3.3 Feature engineering

Our system employs several types of features based on morphological attributes of the messages, POS tags, and the lexica of Section 4.3.2.[9] Both classifiers (Fig. 4.3) use the same features, unless otherwise noted below.

4.3.3.1 Morphological features

- A Boolean feature indicating the existence (or absence) of elongated words (e.g., 'baaad') in the message being classified.
- The number of elongated tokens in the message.
- The existence (or absence) of date expressions in the message (Boolean feature).
- The existence of time expressions (Boolean feature).
- The number of tokens of the message that are fully capitalized (i.e., contain only upper case letters).
- The number of tokens that are partially capitalized (i.e., contain both upper and lower case letters).
- The number of tokens that start with an upper case letter.
- The number of exclamation marks in the message.
- The number of question marks.
- The sum of exclamation and question marks.

[9] All the features are normalized to $[-1, 1]$. In preliminary experiments, we also included as a feature the output of a vagueness classifier (Alexopoulos and Pavlopoulos, 2014), the idea being that vagueness correlates with subjectivity. We do not discuss this feature here, however, since additional experiments are needed to study the correlation of vagueness and subjectivity. We intend to study this issue in future work.

- The number of tokens containing only exclamation marks.
- The number of tokens containing only question marks.
- The number of tokens containing only exclamation or question marks.
- The number of tokens containing only ellipsis (...).
- The existence of a subjective (i.e., positive or negative) emoticon at the message's end.
- The existence of an ellipsis and a link (URL) at the message's end. News tweets, which are often objective, often contain links of this form.
- The existence of an exclamation mark at the message's end.
- The existence of a question mark at the message's end.
- The existence of a question or an exclamation mark at the message's end.
- The existence of slang, as detected by using the slang dictionary (Section 4.3.1).

4.3.3.2 POS based features

- The number of adjectives in the message being classified.
- The number of adverbs.
- The number of interjections (e.g., 'hi', 'bye', 'wow', etc.).
- The number of verbs.
- The number of nouns.
- The number of proper nouns.
- The number of URLs.

– The number of subjective emoticons (for subjectivity detection only).

– The number of positive emoticons (for polarity detection only).

– The number of negative emoticons (for polarity detection only).

– The average, maximum and minimum F_1 scores defined below of the message's POS-tag bigrams for the subjective and neutral classes (for subjectivity detection).

– The average, maximum, and minimum F_1 scores of the message's POS-tag bigrams for the positive and negative classes (for polarity detection only).

For a POS-tag bigram b and a class c, F_1 is calculated over all the training messages as:

$$F_1(b,c) = \frac{2 \cdot Pre(b,c) \cdot Rec(b,c)}{Pre(b,c) + Rec(b,c)} \tag{4.1}$$

where:

$$Pre(b,c) = \frac{\text{\#messages of } c \text{ containing } b}{\text{\#messages containing } b} \tag{4.2}$$

$$Rec(b,c) = \frac{\text{\#messages of } c \text{ containing } b}{\text{\#messages of } c} \tag{4.3}$$

4.3.3.3 Sentiment lexicon based features

For each lexicon of Section 4.3.1, we use seven different features based on the scores provided by the lexicon for each word present in the message.[10]

– Sum of the scores.

– Maximum of the scores.

[10]We removed from SENTIWORDNET any instances having positive and negative scores equal to zero. Moreover, the MPQA lexicon does not provide scores, so, for each word in the lexicon we assume a score equal to 1.

– Minimum of the scores.

– Average of the scores.

– The count of the words of the message that appear in the lexicon.

– The score of the last word of the message that appears in the lexicon.

– The score of the last word of the message.

If a word does not appear in the lexicon, it is assigned a score of 0 and it is not considered in the calculation of the average, maximum, minimum and count scores.

We also created features based on the precision and F_1 scores of the words of MPQA and the words of the lexica generated from the training data (Malakasiotis et al., 2013). For each word w of each lexicon, we calculate the precision ($Pre(w,c)$), recall ($Rec(w,c)$) and F_1 ($F_1(w,c)$) of w with respect to each class c. In the first year of the Task, we used the positive, negative, and neutral classes, but in the second year the subjective class was added in order for features to be computed in two stages. Equations 4.4, 4.5 and 4.6 below provide the definitions of $Pre(w,c)$, $Rec(w,c)$, and $F_1(w,c)$; these measures are computed by counting over the training messages.

$$Pre(w,c) = \frac{\text{\#messages that contain word } w \text{ and belong in class } c}{\text{\#messages that contain word } w} \tag{4.4}$$

$$Rec(w,c) = \frac{\text{\#messages that contain word } w \text{ and belong in class } c}{\text{\#messages that belong in class } c} \tag{4.5}$$

$$F_1(w,c) = \frac{2 \cdot P(w,c) \cdot R(w,c)}{P(w,c) + R(w,c)} \tag{4.6}$$

Having assigned a precision and F_1 score to each word of each lexicon (MPQA and lexica generated from training data), we then compute the sum, maximum, minimum, etc. (as above) of the precision or F_1 scores (separately for each class c) of the words

of the message being classified, and we use them as features of the message being classified.

4.3.3.4 Miscellaneous features

Negation. Not only is negation a good subjectivity indicator, but it also may change the polarity of a message. We therefore add seven more features, one indicating the existence of negation in the message being classified, and the remaining six indicating the existence of negation in the message before (up to a distance of 5 tokens) any words from lexica $S_{\pm}$, S_{+}, S_{-}, $W_{\pm}$, W_{+} and W_{-}. The features involving $S_{\pm}$ and $W_{\pm}$ are used in subjectivity detection only and the remaining four in polarity detection only. We have not implemented the six features for other lexica, but they might be a good addition to the system.

CMU's Twitter clusters. Owoputi et al. (2013) released a dataset of 938 clusters containing words coming from tweets. Words of the same clusters share similar attributes (e.g., they may be near-synonyms, or they may be used in similar contexts). We exploit these clusters by adding 938 Boolean features, each one indicating if any of the message's words appear (or not) in the corresponding cluster.

4.3.4 Feature selection

For feature selection, we first merged the training and development data of SEMEVAL 2013 Task 2 (Section 4.2). Then, we ranked the features with respect to their information gain (Quinlan, 1986) on this merged dataset. To obtain the 'best' set of features, we started with a set containing the top 50 features and we kept adding batches of 50 features until we had added all of them. At each step, we evaluated the corresponding feature set on the TW_{13} and SMS_{13} datasets (Section 4.2). We eventually chose the feature set with the best performance. This resulted in a system which used the top 900

features for Stage 1 and the top 1150 features for Stage 2.

4.4 Experimental Results

The official measure of the SEMEVAL task addressed in this chapter is the average F_1 score of the positive and negative classes ($F_1(\pm)$).[11] Recall that in the first year there were two test sets; one with messages from Twitter and one with SMS messages. On the Twitter test messages, our 2013 system achieved an $F_1(\pm)$ score of 58.91% and it was ranked 13th (out of 36 systems); the best system achieved 69.02% and a majority baseline achieved 29.19%. On the SMS test messages, our 2013 system achieved a score of 55.28% and it was ranked 4th; the best system achieved 68.46% and the baseline 19.03%.

In the second year, our system was improved in three ways. Firstly, we added many more lexica (HL, SENTIWORDNET, AFINN, and NRC). Secondly, we focused on each stage separately. In detail, in the first year, our features were based on three classes, positive, negative, and neutral (Malakasiotis et al., 2013), while in 2014 our features were computed both for the two classes of the first stage (i.e., subjective and neutral) and the two classes of the second second stage (i.e., positive and negative). Thirdly, we removed features based on bag of words (i.e., features showing the presence of specific words or terms), since preliminary feature selection indicated that they did not add much to the other features.

Table 4.1 illustrates the $F_1(\pm)$ score achieved by our 2014 system per evaluation dataset, along with the median and best $F_1(\pm)$. In the same table, AVG_{all} corresponds to the average $F_1(\pm)$ across the five datasets, while AVG_{14} corresponds to the average $F_1(\pm)$ across the 2014 test datasets, i.e., LJ_{14}, TW_{14} and TWSARC_{14}. In all cases our

[11] As noted by the SEMEVAL organisers, this measure does not make the task binary. In effect, the neutral class is considered less important than (and is being indirectly evaluated through) the positive and negative classes.

Test Set	**AUEB**	**Median**	**Best**
LJ_{14}	70.75	65.48	74.84
SMS_{13} *	64.32	57.53	70.28
TW_{13} *	63.92	62.88	72.12
TW_{14}	66.38	63.03	70.96
$TWSARC_{14}$	56.16	45.77	58.16
AVG_{all}	64.31	56.56	68.78
AVG_{14}	64.43	57.97	67.62

Table 4.1: $F_1(\pm)$ scores of our 2014 system per dataset. Stars indicate datasets that were also used for feature selection.

Test Set	**Ranking**
LJ_{14}	9/50
SMS_{13} *	8/50
TW_{13} *	21/50
TW_{14}	14/50
$TWSARC_{14}$	4/50
AVG_{all}	6/50
AVG_{14}	5/50

Table 4.2: Rankings of our system among SEMEVAL-2014 participants. Stars indicate datasets that were also used for feature selection.

results are above the median. Table 4.2 illustrates the ranking of our system according to $F_1(\pm)$. Our system ranked 6th according to AVG_{all} and 5th according to AVG_{14} among the 50 participating systems of the 2014 competition. Recall that in 2014 the test data comprised tweets (TW_{14}), tweets including sarcasm (TWSARC_{14}), SMS messages, and sentences from LIVEJOURNAL.

4.5 Conclusions and future work

In this chapter, we presented a system we designed and implemented for message-level sentiment estimation. Our system participated, and was highly ranked, in the Message Polarity Classification subtask of the Sentiment Analysis in Twitter Task of SEMEVAL 2013 and 2014. We proposed a two–stage pipeline approach, which first detects sentiment (objective vs. subjective messages) and then decides about the polarity of the message, using two separate SVM classifiers. The results indicate that our system handles well the class imbalance problem and has a good generalization ability over different types of messages (tweets, SMSs, blog posts from LIVEJOURNAL). The next chapter discusses how the system of this chapter can be used in ABSA, where we need to determine the sentiment expressed for each mentioned aspect term (or coarser aspect) of a target entity.

Chapter 5

Aspect Term Sentiment Estimation

5.1 Introduction

In this chapter we focus on estimating the sentiment for each aspect term of a target entity. Given a set of sentences from reviews of the target entity (e.g., reviews of a laptop or restaurant), we assume that all the aspect terms have been correctly detected (e.g., by an aspect term extraction method discussed in Chapter 2), and we aim to determine the sentiment polarities of the aspect term occurrences of each sentence. This was also a subtask called 'Aspect term polarity' of the task of 'Aspect Based Sentiment Analysis' (ABSA) in SEMEVAL 2014.[1] As in SEMEVAL 2014, we assume that the polarity of an aspect term occurrence can be positive, negative, neutral (i.e., neither positive nor negative) or conflict (i.e., both positive and negative).[2] For example:

- In "I loved their fajitas", 'fajitas' has positive polarity.
- In "I hated their fajitas, but their salads were great", 'fajitas' has negative polarity while 'salads' has positive polarity.

[1] Consult `http://alt.qcri.org/semeval2014/task4/`.

[2] Instead of conflict, we could look for the dominant sentiment, but then it would be more difficult for human judges to agree on the correct polarities of the aspect terms.

- In "The fajitas were their starters", 'fajitas' has neutral polarity.
- In "The fajitas were great to taste, but not to see", 'fajitas' has conflict polarity.

As will be discussed, two of our datasets (restaurants, laptops) were validated and extended by expert human annotators in order to be used as benchmarks in the 'Aspect term polarity' subtask of the SEMEVAL 2014 ABSA task. We also discuss and report the results of the SEMEVAL 2014 ABSA subtask.

Message-level sentiment estimation systems, such as the one described in Chapter 4, can be directly applied to aspect term sentiment estimation by classifying each sentence (or part of the sentence) containing an aspect term occurrence. Then, all the aspect term occurrences of the sentence are assigned the sentiment polarity returned by the system (i.e., the polarity of the sentence). This approach, however, is problematic in sentences that contain aspect term occurrences with different polarities, as in "The steak was good but awful service", where the aspect term 'steak' has positive polarity while 'service' has negative polarity. We will show that sentences with different polarities are relatively rare in the datasets we considered and, thus, the application of message-level systems to aspect term sentiment estimation can be effective, at least in the types of reviews we experimented with. Consequently, we applied our message-level sentiment estimation system of Section 4.3 to aspect term sentiment estimation datasets and we discuss its performance.

5.2 Aspect term polarity datasets

The human annotators that annotated the aspect terms of our aspect term extraction datasets (Section 2.2) also assigned a sentiment polarity label (positive, negative, neutral, or conflict) to each aspect term occurrence. Our aspect term polarity datasets comprise the sentences of our aspect term extraction datasets along with the gold (human) polarity label of each aspect term occurrence.

```
<sentence id="11351725#582163#9">
   <text>Our waiter was friendly and it is a shame that he didn't
   have a supportive staff to work with.</text>
        <aspectTerms>
            <aspectTerm term="waiter" polarity="positive" from="4"
            to="10"/>
            <aspectTerm term="staff" polarity="negative" from="74"
            to="79"/>
        </aspectTerms>
        <aspectCategories>
            <aspectCategory category="service" polarity="conflict"/>
        </aspectCategories>
</sentence>
```

Figure 5.1: XML annotation of a sentence from the restaurants dataset of the SEMEVAL 2014 ABSA task.

To measure inter-annotator agreement, we used the same sample of 75 restaurant, 75 laptop, 100 hotel sentences and the same annotators as in aspect term extraction (Section 2.2). Agreement was measured using Cohen's Kappa (Cohen, 1960) by considering the sentiment classes the annotators assigned to their common (tagged by both annotators) aspect term occurrences. We obtained $K = 75.34, 87.25, 74.95\%$, respectively, for restaurants, hotels, and laptops, i.e., substantial agreement; for hotels, we report the mean pairwise K, since there were three annotators in this agreement study. Additional restaurant and laptop reviews, were also annotated in the same manner by the expert annotators if the SEMEVAL 2014 ABSA task to be used as test data.

More precisely, from our initial laptops dataset, 3045 sentences were used as training data in the SEMEVAL 2014 ABSA task and 800 as test data. From our initial restaurants dataset, 3041 sentences were used as training data in the SEMEVAL 2014 ABSA task and 800 new sentences were collected, annotated, and used as test data. The an-

	Positive	Negative	Neutral	Conflict	Total
Train	2164	805	633	91	3693
Test	728	196	196	14	1134
Total	2892	1001	829	105	4827

Table 5.1: Class statistics for the restaurants dataset.

	Positive	Negative	Neutral	Conflict	Total
Train	987	866	460	45	2358
Test	341	128	169	16	654
Total	1328	994	629	61	3012

Table 5.2: Class statistics for the laptops dataset.

notation guidelines that were provided to the expert human annotators are available online.[3] Any disagreements between the human experts were resolved between them or with the help of a third judge when required (e.g., when the judges could not reach a consensus). Consult Pontiki et al. (2014) for more information on the types of disagreements between the annotators. The sentences of the datasets were provided to the participants of the ABSA task of SEMEVAL 2014 in an XML format (Fig. 5.1).

In the reminder of this chapter, we use the aspect term polarity datasets of the ABSA task of SEMEVAL 2014, as they were corrected and modified by the expert annotators. The most common sentiment polarity in both domains (restaurants, laptops) is 'positive'. Tables 5.1 and 5.2 provide statistics about these datasets and the distribution of the sentiment classes.

[3] See `http://alt.qcri.org/semeval2014/task4/data/uploads/semeval14_absa_annotationguidelines.pdf`.

5.3 Aspect term polarity evaluation measures

Aspect term polarity methods can be evaluated using *accuracy* (*Acc*), defined as the number of correctly classified (as positive, negative, neutral, or conflict) aspect term occurrences divided by the total number of aspect term occurrences. This was also the official measure of the 'Aspect term polarity' subtask of the ABSA task of SEMEVAL 2014.

Although *Acc* is a well established measure, it is also well known that when it is applied to skewed class distributions it can lead to misleading results. Instead, one can compute precision, recall, and F_1-measure per class. In our case, *positive precision* (P_+) is the number of aspect term occurrences correctly classified as positive divided by the total number of aspect term occurrences classified as positive. *Positive recall* (R_+) is the number of aspect term occurrences correctly classified as positive divided by the total number of truly positive aspect term occurrences. The precision and recall of the other classes, i.e., conflict (c) and neutral (n) are defined similarly, and $F_\kappa = F_{\beta=1,\kappa} = 2 \cdot \frac{P_\kappa \cdot R_\kappa}{P_\kappa + R_\kappa}$, for $\kappa \in \{+,-,c,n\}$.

Precision, recall, and F-measure, however, do not take into account that the positive class is closer to the neutral and conflict ones than to the negative class. For example, classifying a positive aspect term occurrence as negative should be penalized more heavily than classifying it as neutral or conflict. Also, only average polarity scores (over all the reviews of the target entity) are actually needed in ABSA systems; classification errors not affecting the average polarity scores of the aspect terms (the stars of Fig. 2.1) do not actually matter. Furthermore, it is more important to compute accurately the average polarity scores of the most frequent aspect terms, because less frequent aspect terms will not be shown to end-users (there will be no rows for them in Fig. 2.1).

To address these issues, we propose using the *mean absolute error*. Let v_i be the *predicted* (by a method being evaluated) average polarity of the distinct aspect term a_i, counting each predicted positive occurrence of a_i as $+1$, each predicted negative

occurrence as -1, and each predicted neutral or conflict occurrence as 0.[4] Similarly, let v_i^* be the *true* average polarity of a_i, i.e., the average polarity that is based on the classes (polarity labels) assigned to the occurrences of a_i by the human annotators, again counting each positive occurrence as $+1$, each negative occurrence as -1, and each neutral or conflict occurrence as 0. Both v_i and v_i^* range in $[-1,+1]$, hence $\frac{1}{2} \cdot |v_i - v_i^*|$ ranges in $[0,1]$. The mean absolute error MAE_m of an aspect term polarity estimation method for the m most frequent distinct aspect terms is defined as:

$$MAE_m = \frac{1}{2m} \cdot \sum_{i=1}^{m} |v_i - v_i^*|$$

To compare two methods, one can plot their MAE_m for $m = 1, \ldots, |G|$, where $|G|$ is the number of gold distinct aspect terms. Alternatively one can compute MAE_m for particular indicative values of m (e.g., $m = 5, 10, 50, |G|$). We also describe below a data exploratory analysis, which can be used to obtain prominent values of m per domain.

5.4 Using the system of Chapter 4 to estimate aspect term polarities

Message-level sentiment estimation systems, such as our system of Chapter 4, can be used for aspect term polarity estimation if all the aspect term occurrences in each sentence are assigned the sentiment polarity returned by the system for the entire sentence. As shown in Figures 5.2 and 5.3, most *multi-aspect sentences* (meaning sentences containing more than one aspect terms) of the datasets used in this chapter contain aspect terms which have been annotated by the human annotators with the same sentiment polarity. This means that a message-level sentiment estimation system, which assigns the same polarity to all the aspect terms of each sentence, can also be effective. Hence,

[4] In the prototype of Fig. 2.1, the v_i scores are mapped to a 1–5 scale, and they are displayed as one to five stars.

we applied our system of Chapter 4 to each sentence of the two test datasets (laptops, restaurants) of the 'Aspect term polarity' subtask of the ABSA task of SEMEVAL 2014, and we tagged each aspect term occurrence of each sentence with the polarity (positive, negative, neutral) returned by our system for the entire sentence.[5] We stress that we performed no extra training, i.e., our system was trained on tweets of no particular domain, as in Chapter 4. We thus consider our system to be *domain-independent*.

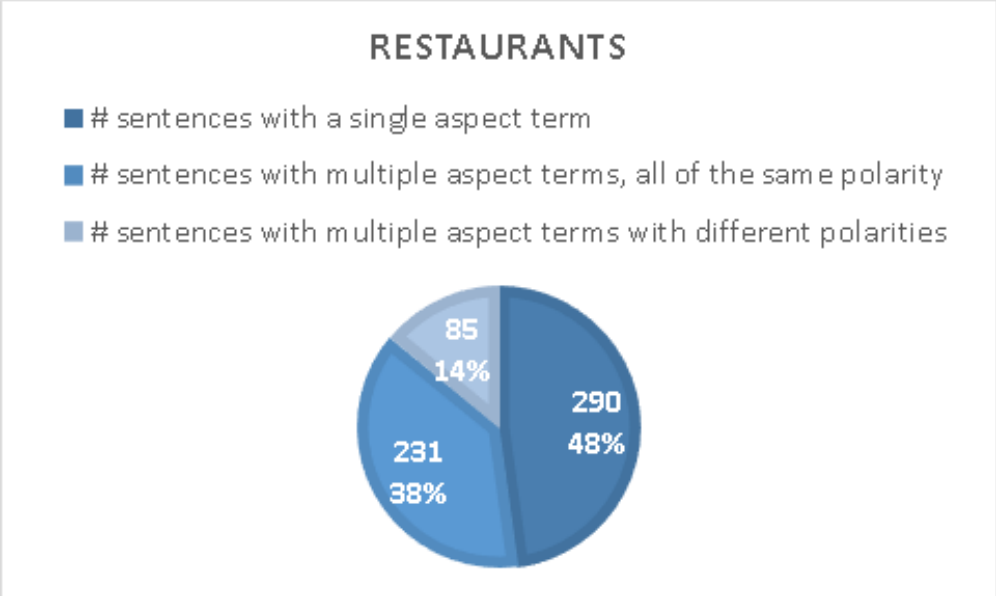


Figure 5.2: Sentences of the restaurants dataset containing (one or more) aspect terms annotated with a single polarity or more than one polarities.

[5] We never assign a conflict polarity to any aspect term, since our system of Chapter 4 never assigns a conflict polarity to a sentence.

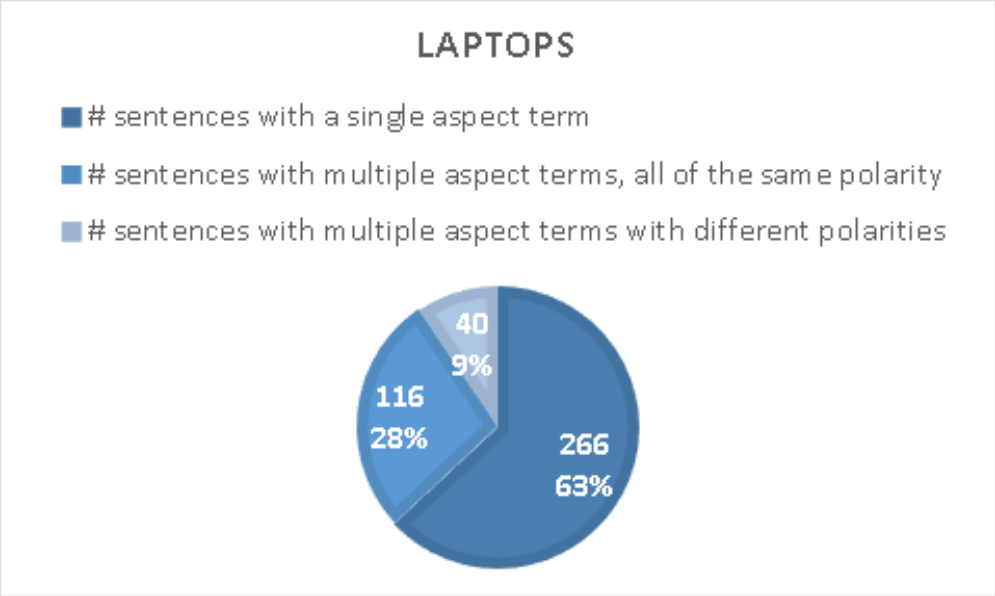


Figure 5.3: Sentences of the laptops dataset containing (one or more) aspect terms annotated with a single polarity or more than one polarities.

5.5 Aspect term polarity results and comparison to SEMEVAL systems

The 'Aspect term polarity' subtask of ABSA of SEMEVAL 2014 attracted 26 teams for the laptops dataset, and 26 teams for the restaurants dataset. We briefly present here the official baseline of the subtask and the best (based on their *Acc* scores) systems that participated in the subtask, relying on information provided by Pontiki et al. (2014). We also report the evaluation results of our system, the baseline, and the best competitor systems.

For each aspect term t in a test sentence s (of a particular domain), the baseline checks if t had been encountered in the training sentences (of the domain). If so, it retrieves the k most similar to s training sentences (of the domain), and assigns to the aspect term t the most frequent polarity it had in the k sentences. Otherwise, if t had not

been encountered in the training sentences, it is assigned the most frequent aspect term polarity label of the training set. The similarity between two sentences is measured as the Dice coefficient D of the sets of (distinct) words of the two sentences. For example, the similarity between "this is a demo" (s_1 and "that is yet another demo" (s_2) is: $D(s_1, s_2) = \frac{2 \cdot |s_1 \cap s_2|}{|s_1| + |s_2|} = \frac{2 \cdot 2}{4+5} = 0.44$. Notice that the baseline is supervised and domain-dependent, since it relies on the training data of a particular domain. By contrast, our system is domain-independent, as already discussed.

We also describe the systems of the DCU and NRC-Canada teams, which were the best in both domains. Both NRC-Canada and DCU relied on an SVM classifier with features based on n-grams and several domain-independent, publicly available sentiment lexica (e.g., MPQA, SentiWordnet and Bing Liu's Opinion Lexicon). NRC-Canada also used features based on POS-tags and parse trees, as well as two automatically compiled polarity lexica for restaurants and laptops, obtained from YELP and Amazon data, respectively. Furthermore, NRC-Canada showed by ablation experiments that the most useful features are those derived from the sentiment lexica. On the other hand, DCU used only publicly available lexica, which were manually adapted by filtering words that do not express sentiment in laptop and restaurant reviews (e.g., 'really') and by adding others that were missing and do express sentiment (e.g., 'mouthwatering'). As was the official baseline described above, these two systems are supervised and domain-dependent.

Our system achieved 0.427 error rate (which is 1-*Acc*) in laptops and 0.318 in restaurants, which ranks us 16th out of 26 participating teams in laptops and 16th out of 26 participating teams in restaurants.[6] DCU and NRC-Canada, which had the best systems in both domains, achieved identical scores (0.295 error rate) on the laptops dataset (Table 5.4) while the DCU system performed slightly better (0.191 vs. 0.199 error rate) on

[6]Teams participating in the 'Aspect term polarity' subtask of the ABSA task of SEMEVAL 2014 were allowed to submit more than one systems. We consider only the best performing system for each team.

Teams	**Error rate (1-Acc)**	$MAE_{m=50}$	$MAE_{m=500}$
DCU	0.191 (1st/26)	0.076 (2nd/26)	0.126 (1st/26)
NRC	0.199 (2nd/26)	0.062 (1st/26)	0.141 (2nd/26)
XRCE	0.223 (3rd/26)	0.088 (4th/26)	0.156 (4th/26)
UWB	0.223 (4th/26)	0.090 (5th/26)	0.143 (3rd/26)
SZTENLP	0.248 (5th/26)	0.120 (10th/26)	0.164 (5th/26)
AUEB	0.318 (16th/26)	0.097 (6th/26)	0.194 (8th/26)
Baseline	0.357 (22th/26)	0.769 (26th/26)	0.737 (24th/26)

Table 5.3: Error rate (1-*Acc*), $MAE_{m=50}$, and $MAE_{m=500}$ of our system and the top-5 systems of the aspect term polarity subtask of SEMEVAL 2014, in the restaurants domain.

Teams	**Error rate (1-Acc)**	$MAE_{m=50}$	$MAE_{m=500}$
DCU	0.295 (1st/26)	0.118 (3rd/26)	0.165 (3rd/26)
NRC	0.295 (2nd/26)	0.141 (12th/26)	0.160 (2nd/26)
IITPatan	0.330 (3rd/26)	0.111 (1st/26)	0.178 (4th/26)
SZTENLP	0.330 (4th/26)	0.125 (4th/26)	0.190 (7th/26)
UWB	0.333 (5th/26)	0.118 (2nd/26)	0.182 (5th/26)
AUEB	0.427 (16th/26)	0.147 (13th/26)	0.201 (11th/26)
Baseline	0.486 (25th/26)	0.704 (25th/26)	0.687 (25th/26)

Table 5.4: Error rate (1-*Acc*), $MAE_{m=50}$, and $MAE_{m=500}$ of our system and the top-5 systems of the aspect term polarity subtask of SEMEVAL 2014, in the laptops domain. The lower the scores, the better the system

the restaurants dataset (Table 5.4).

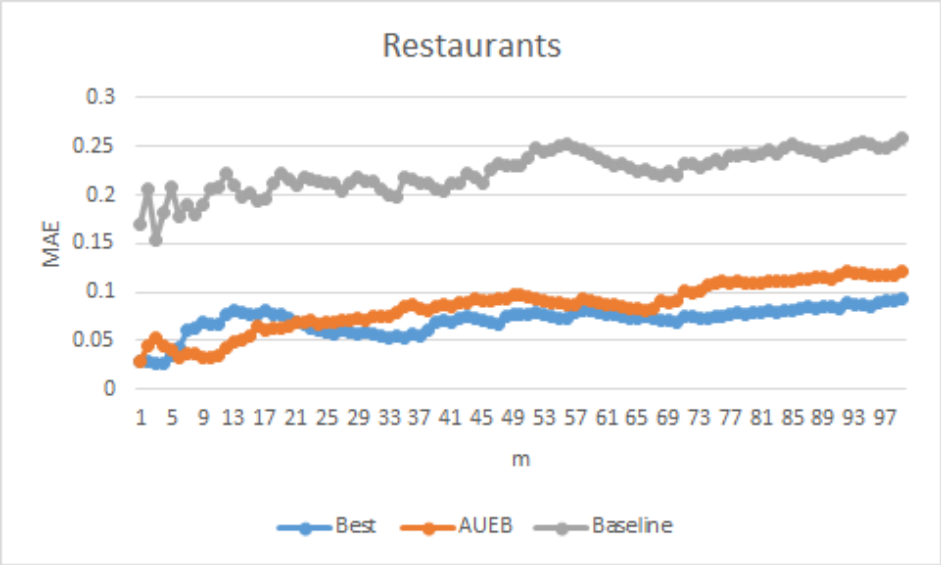


Figure 5.4: MAE_m scores, for m ranging from 1 to 100, for our system, a baseline and the best system of the ABSA task of SEMEVAL 2014, in the restaurants domain.

Figures 5.4 and 5.5 show the MAE_m (for m ranging from 1 to 100) of our system, the baseline, and the best system of the 'Aspect term polarity' subtask of SEMEVAL 2014. We see that our system is clearly better than the baseline (the lower the error the better the system) and worse than the best system. However, we observe that our system is very close to the best system in the restaurants dataset, and also in the laptops dataset for $m \geq 50$. This is particularly important considering that our system was used directly as it was trained (Chapter 4) on tweets of no particular domain.

In Tables 5.3 and 5.4, we show the error rate (1-Acc) score of our system of Chapter 4, again applied to the test data of the aspect term polarity subtask of SEMEVAL 2014, along with the scores of the top-5 systems.[7] Also, for each system, we show its MAE_m score for two indicative values of m ($m = 50$ and $m = 500$). We chose $m = 50$ based on the cumulative frequency of aspect term occurrences per number of distinct

[7]For teams that submitted more than one systems, the best performing one was chosen.

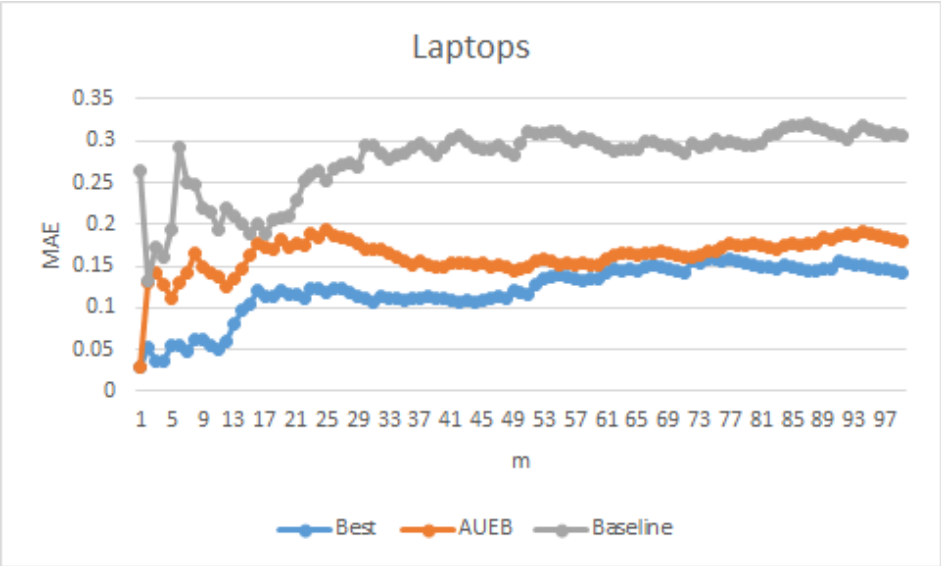


Figure 5.5: MAE_m scores, for m ranging from 1 to 100, for our system, a baseline and the best system of the ABSA task of SEMEVAL 2014, in the laptops domain.

aspect terms. As shown in Figures 5.6 and 5.7, more than half of the aspect term occurrences stem from only 50 (most) frequently annotated distinct aspect terms. Furthermore, each of other aspect terms affect less sharply the cumulative distribution. Also, we use $m = 500$, which is a value in between the 50 frequent and all possible distinct aspects.

Our system achieves better rankings with $MAE_{m=50}$ and $MAE_{m=500}$ than when evaluated with the error rate. Also, we observe that the MAE_m scores of all the systems are higher for $m = 500$ and relatively closer to the error rates than when $m = 50$. Also, for $m = 50$ there are bigger changes in the system rankings obtained via MAE_m compared to the rankings obtained via error rate. These observations are due to (i) the fact that MAE_m considers only the occurrences of the m most frequent distinct aspect terms, unlike error rate, and (ii) it also takes into account the fact that the positive class is closer to the neutral and conflict classes than to the negative one. For larger m values, difference (i) becomes less intense, but difference (ii) remains.

Figure 5.6: Cumulative frequency of aspect term occurrences per number of distinct aspect terms, in the restaurants domain.

5.6 Experiments with ensembles

We also studied the contribution of our system when it is adopted in an ensemble classifier. We developed a classifier that uses the responses of the two best systems (NRC and DCU) of the aspect term polarity subtask of SEMEVAL 2014, plus the responses of one more system, in order to return the polarity of an aspect term. In a first set of experiments, for each aspect term, the three systems vote; i.e., if any two systems agree, their response is returned and if they disagree, the ensemble returns the response of the best system (DCU). In a second set of experiments, a voting is performed only for the aspect terms where the two best systems disagree; if the two best systems agree, the ensemble returns their decision without consulting the third system. We call EC1 and EC2 the ensemble of the first and second set of experiments, respectively. In both EC1 and EC2, we experimented with both our system and the UWB system as the third

Ensemble classifier	Error rate (1-Acc)	$MAE_{m=50}$
EC1-UWB	0.198	0.081
EC2-UWB	**0.184**	0.075
EC1-AUEB	0.202	0.070
EC2-AUEB	0.196	**0.058**
Best	0.190	0.076

Table 5.5: Error rate (1-*Acc*) and $MAE_{m=50}$ of four ensemble classifiers and the best SEMEVAL participant in the **restaurants** domain.

Ensemble classifier	Error rate (1-Acc)	$MAE_{m=50}$
EC1-UWB	0.283	**0.100**
EC2-UWB	0.282	0.120
EC1-AUEB	**0.269**	0.114
EC2-AUEB	0.282	0.108
Best	0.295	0.118

Table 5.6: Error rate (1-*Acc*) and $MAE_{m=50}$ of four ensemble classifiers and the best participant of the subtask in the **laptops** domain.

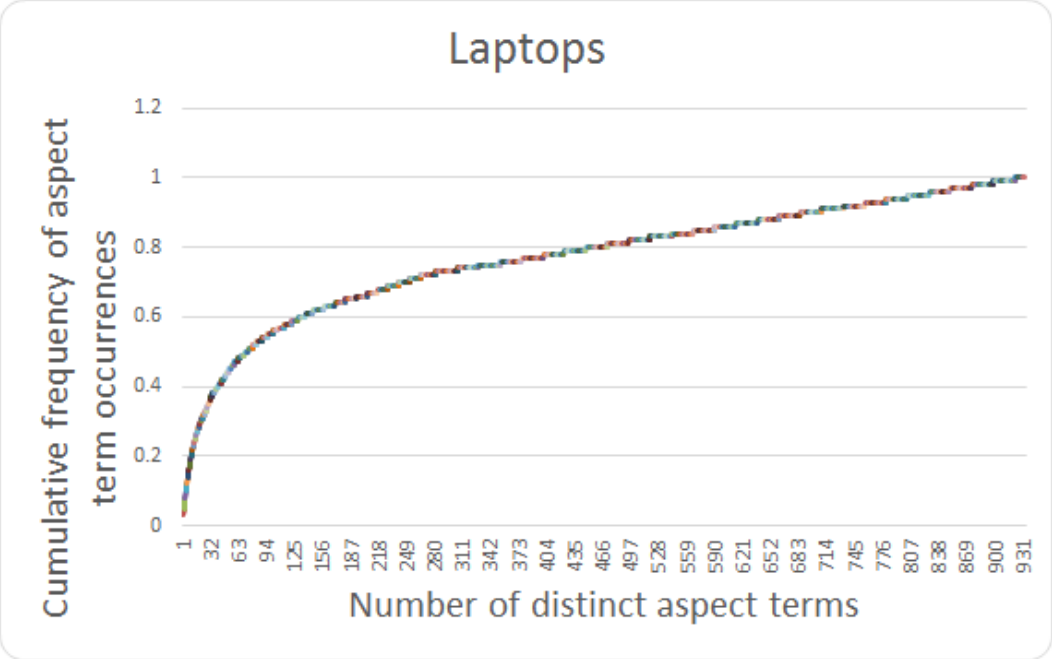


Figure 5.7: Cumulative frequency of aspect term occurrences per number of distinct aspect terms, in the laptops domain.

system of the ensemble. We used the UWB system, because it was ranked within the five best systems in both domains.[8] We denote EC1-UWB and EC1-AUEB the EC1 ensemble with UWB or our system respectively, as the third system of the enselmbe, and similarly for EC2-UWB and EC2-AUEB.

Tables 5.5 and 5.6 show the error rate (1-*Acc*) and $MAE_{m=50}$ scores of the four ensembles. Overall, we see that the ensemble classifier has the potential to be the best system of the subtask. In restaurants, EC2-UWB achieves better error rate than the best participant of the subtask. In laptops, all the ensembles achieve better error rate scores than the best participant in the subtask. When we evaluate with $MAE_{m=50}$, both EC1-AUEB and EC2-AUEB achieve better scores than the best participant of the subtask. Also, EC2-AUEB, in restaurants, achieves the best score among all four ensembles.

[8]The systems XRCE and IITPatan were not in the top-5 in both domains, while SZTENLP was ranked in the same positions as UWB (we made our choice for UWB arbitrarily).

Hence, the contribution of integrating our system into an ensemble classifier may be, in effect, greater than integrating a highly ranked system, such as UWB.

5.7 Other related work on aspect term polarity

Prior to the ABSA task of SEMEVAL 2014, most previous work on sentiment polarity was concerned with assigning sentiment labels or scores to entire texts (Liu, 2012; Pang and Lee, 2004; Pang and Lee, 2008), sentences (Turney, 2002; Kim and Hovy, 2004), tweets (Davidov et al., 2010), or syntactic constituents (Wilson et al., 2005; Socher et al., 2013). By contrast, the goal in this chapter is to estimate the sentiment polarity of particular given aspect terms. We follow Hu and Liu (2004) and require a polarity score for each *occurrence* of an aspect term. The polarity scores of all the occurrences of an aspect term, however, can then be averaged over all the input texts. If an aspect aggregation stage is also present (Chapter 3), the polarity scores of aspect terms that have been clustered together can also be averaged (Fig. 2.1).

In terms of evaluation measures, Hu and Liu (2004) measure the accuracy of classifying aspect term occurrences into two polarity classes (positive and negative). Moghaddam and Ester (2010) rank the (distinct) aspect terms by their polarity scores and use a ranking loss coefficient (Snyder and Barzilay, 2007) to measure the average distance between the true and the predicted rankings. Ranking-based measures of this kind are appropriate for methods that attempt to rank the (distinct) aspect terms of a target entity by opinion (e.g., from the most positive to the most negative aspect terms), without estimating exactly how positive or negative the opinions are, and without assigning polarity labels to term occurrences, unlike our aspect term polarity estimation subtask. Mean average error has been used in previous sentiment polarity work (Baccianella et al., 2009; Long et al., 2010), but for the polarity of entire documents, not aspect terms, and not as a function of m.

5.8 Conclusions

In this chapter we discussed the task of aspect term sentiment estimation, which was also a subtask of the ABSA task of SEMEVAL 2014. We described the highest ranked systems that participated in the SEMEVAL subtask and we showed that message-based sentiment estimation systems can also be applied to the subtask. We applied our domain-independent message-based system, discussed in Chapter 4, and we showed that it can be reasonably effective without retraining. Also, we proposed an evaluation measure which is more adequate for the subtask and we studied the performance of systems that participated in the 'Aspect term polarity' subtask of SEMEVAL 2014, when evaluated with our proposed measure. Four ensemble classifier were also developed, that use the responses of three systems and we showed that when our system is integrated in the ensemble, and we use our evaluation measure, the ensemble classifier outperforms all the individual systems that participated in SEMEVAL subtask.

Chapter 6

Conclusions

6.1 Summary and contribution of this thesis

Aspect Based Sentiment Analysis (ABSA) systems receive as input a set of texts (e.g., product reviews or messages from social media) discussing a particular entity (e.g., a new model of a mobile phone). The systems attempt to detect the main (e.g., the most frequently discussed) aspects (features) of the entity (e.g., 'battery', 'screen') and to estimate the average sentiment of the texts per aspect (e.g., how positive or negative the opinions are on average for each aspect). Although several ABSA systems have been proposed, mostly research prototypes (Liu, 2012), there is no established task decomposition for ABSA, nore are there any established evaluation measures for the subtasks ABSA systems are required to perform.

This thesis, proposed a new task decomposition for ABSA, which contains three main subtasks: aspect term extraction, aspect term aggregation, and aspect term polarity estimation. The first subtask detects single- and multi-word terms naming aspects of the entity being discussed. The second subtask aggregates similar aspect terms. The third subtask estimates the average sentiment per aspect term or cluster of aspect terms.

For each one of the above mentioned subtasks, benchmark datasets for different

kinds of entities were constructed during the work of this thesis. New evaluation measures were introduced for each subtask, arguing that they are more appropriate than previous evaluation measures. For each subtask, the thesis proposed new methods (or improvements over previous methods) and it was shown experimentally on the constructed benchmark datasets that the new methods (or the improved versions) are better or at least comparable to state of the art ones.

More specifically, for the aspect term extraction (ATE) subtask, new benchmark datasets were constructed, which have also been adopted (with some changes) by an international challenge (the ABSA Task of SEMEVAL 2014 and 2015) coorganized by the author. Also, it was shown that there is reasonable inter-annotator agreement, when humans are asked to annotate aspect terms in texts. The thesis introduced new weighted variants of precision, recall, and average precision, explaining why the new variants are better than the standard ones when evaluating ATE methods. The thesis also proposed an improved form of a popular unsupervised ATE method (Hu and Liu, 2004), where an extra pruning stage that removes candidate aspect terms (based on recently popular methods that map words to continuous space vectors) is added. Lastly, the thesis showed experimentally, using the introduced datasets and evaluation measures, that the new improved method, with the extra pruning stage, is significantly better than the original method.

In the aspect aggregation subtask of ABSA, the thesis introduced the problem of aspect aggregation at multiple granularities and proposed decomposing the problem in two phases. In the first phase, systems attempt to fill in a similarity matrix. In the second phase, systems use the generated similarity matrix of the first phase, along with a linkage criterion, and perform hierarchical agglomerative clustering in order to create an aspect term hierarchy; by intersecting the hierarchy at different depths, different numbers of clusters are produced, satisfying different user preferences and other restrictions (e.g., size of screen). The thesis showed experimentally, using aspect ag-

gregation datasets constructed by the author, that the proposed decomposition leads to high inter-annotator agreement and allows re-using existing similarity measures (for the first phase) and hierarchical clustering methods (for the second phase). A novel sense pruning mechanism was also devised, which improves significantly all the existing WordNet-based similarity measures that were tested in the first phase. The experimental results show, however, that there is still large scope for improvements in the methods of the first phase. Lastly, the thesis showed that the second phase is not really affected by the linkage criterion and that it leads to near perfect results (based on human judgments) when a human-generated similarity matrix is used in the first phase.

For aspect term polarity estimation, the author first participated in the development of a system that estimates the sentiment (positive, negative, or neutral) of whole messages (e.g., tweets). The sentiment estimation system participated in international competitions (Sentiment Analysis in Twitter task of SEMEVAL 2013 and 2014) with very good results. The results also showed that the system performed better than most of the other competitors when applied to messages of a different nature than those seen during training (e.g., SMS messages instead of tweets); thus, the system has a very good generalization ability.

The message-level sentiment estimation system of the previous paragraph was then tested on aspect term polarity estimation datasets, constructed during the work of this thesis, where the goal is to estimate the sentiment for each aspect term, rather than for each entire message (e.g., sentence). In this case, the system was used to classify each entire message into a sentiment class (positive, negative, or neutral), and then the sentiment class of the message was also assigned to all of the aspect terms of the message, assuming that all the aspect terms of a message carry the same sentiment. Although there are messages containing aspect terms of different polarities (e.g., one positive and one negative aspect term), an analysis showed that messages of this kind are relatively rare, at least in the datasets of the experiments. Consequently, the system performs rea-

sonably well compared to competing systems, even though it was not re-trained (unlike the competition). A new evaluation measure for the subtask of aspect term polarity estimation was also proposed, which takes into account that (i) misclassifying, for example, a positive aspect term into the negative class is a bigger mistake than misclassifying it into the neutral category; and (ii) that the end users of ABSA systems are interested mainly in frequent aspect terms; mistakes not affecting the average polarity scores of frequent aspect terms do not really matter. With the new evaluation measure, the performance of the message-level system of the previous paragraph is even better compared to its competitors. The ranking of the top systems that participated in the corresponding subtask of an international competition (the ABSA Task of SEMEVAL 2014 and 2015) is also changed, when the new measure is used, as opposed to using the official measures of the competition. The datasets that were created for this subtask during the work of this thesis were also used (with some changes) in the same competition.

6.2 Future work

Future work in the area of ABSA should construct even more benchmark datasets, from a greater variety of domains, to allow safer conclusions to be drawn. Also, it would be interesting to add a summarization (final) step to the ABSA decomposition of this thesis, to show a short summary for each aspect term (or group of aspect terms).

More specifically, in ATE, it would be interesting to compare (only) unsupervised systems that extract aspect terms. Note that, although unsupervised systems are quite popular in ATE, only one unsupervised system participated in the 'Aspect term extraction' subtask of ABSA in SEMEVAL 2014.

In aspect aggregation, cluster labeling would be a very helpful addition. For each group (cluster) of aspect terms, a label could be automatically extracted reflecting the concept of all (or most of) the aspect terms in the group. Also, more systems could be

applied in Phase A (it was shown that Phase B is an almost solved problem) in order to reach the performance of humans in the task. The sense pruning mechanism could be applied in more word similarity tasks as well as examine its applicability to other research problems.

In aspect term sentiment estimation, it will be interesting if a regression system is developed, which, instead of classifying sentences containing an aspect term and averaging the sentiment, will be trained on these data and given a new set of sentences (containing an aspect term) it will be returning a sentiment score.

References

E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa. 2009. A study on similarity and relatedness using distributional and Wordnet-based approaches. In *Proceedings of NAACL*, pages 19–27, Boulder, CO, USA.

E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo. 2013. sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In *The Second Joint Conference on Lexical and Computational Semantics*.

P. Alexopoulos and J. Pavlopoulos. 2014. A vague sense classifier for detecting vague definitions in ontologies. In *Proceedings of EACL*, pages 33–37, Gothenburg, Sweden.

S. Baccianella, A. Esuli, and F. Sebastiani. 2009. Multi-facet rating of product reviews. In *Proceedings of ECIR*, pages 461–472, Toulouse, France.

S. Baccianella, A. Esuli, and F Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of LREC*, Valletta, Malta.

A. Bagheri, M. Saraee, and F. Jong. 2013. An unsupervised aspect detection model for sentiment analysis of reviews. In *Proceedings of NLDB*, volume 7934, pages 140–151. ACL.

L. Barbosa and J. Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of COLING*, pages 36–44, Beijing, China.

D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. 2003a. Hierarchical topic models and the nested Chinese restaurant process. In *Proceedings of NIPS*, pages 17–24, Vancouver, Canada.

D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003b. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

J. Blitzer, M. Dredze, and F. Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL*, pages 440–447, Prague, Czech Republic.

D. Bollegala, Y. Matsuo, and M. Ishizuka. 2007a. An integrated approach to measuring semantic similarity between words using information available on the web. In *Proceedings of HLT-NAACL*, pages 340–347, Rochester, NY, USA.

D. Bollegala, Y. Matsuo, and M. Ishizuka. 2007b. Measuring semantic similarity between words using web search engines. In *Proceedings of WWW*, volume 766, pages 757–766, Banff, Alberta, Canada.

G. Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40.

S. Brody and N. Elhadad. 2010. An unsupervised aspect-sentiment model for online reviews. In *Proceedings of NAACL*, pages 804–812, Los Angeles, CA, USA.

A. Budanitsky and G. Hirst. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

G. Carenini, R. T. Ng, and E. Zwart. 2005. Extracting knowledge from evaluative text. In *Proceedings of K-CAP*, pages 11–18, Banff, Alberta, Canada.

H. Chen, M. Lin, and Y. Wei. 2006. Novel association measures using web search with double checking. In *Proceedings of COLING-ACL*, pages 1009–1016, Sydney, Australia.

P. Cimiano and S. Staab. 2005. Learning concept hierarchies from text with a guided hierarchical clustering algorithm. In *Proceedings of ICML*, Bonn, Germany.

P. Cimiano, A. Mädche, S. Staab, and J. Völker. 2009. Ontology learning. In *Handbook on Ontologies*, pages 245–267. Springer.

J. Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

N. Cristianini and J. Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

D. Davidov, O. Tsur, and A. Rappoport. 2010. Enhanced sentiment learning using Twitter hashtags and smileys. In *Proceedings of COLING*, pages 241–249, Beijing, China.

D. L. Davies and D. W. Bouldin. 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227.

X. Ding, B. Liu, and P. S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of WSDM*, pages 231–240, Palo Alto, CA, USA.

J. C. Dunn. 1974. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

C. Fellbaum, editor. 1998. *WordNet. An Electronic Lexical Database*. MIT Press.

T. Fountain and M. Lapata. 2012. Taxonomy induction using hierarchical random graphs. In *Proceedings of NAACL:HLT*, pages 466–476, Montreal, Canada.

G. Ganu, N. Elhadad, and A. Marian. 2009. Beyond the stars: Improving rating predictions using review text content. In *Proceedings of WebDB*, Providence, RI, USA.

H. Guo, H. Zhu, Z. Guo, X. Zhang, and Z. Su. 2009. Product feature categorization with multilevel latent semantic association. In *Proceedings of CIKM*, pages 1087–1096.

L. Han, A. Kashyap, T. Finin, J. Mayfield, and J. Weese. 2013. Umbc_ebiquity-core: Semantic textual similarity systems. In *Proceedings of SemEval*, pages 44–52, Atlanta, GA, USA.

Z. Harris. 1968. *Mathematical Structures of Language*. Wiley.

T. Hastie, R. Tibshirani, and J. Friedman. 2001. *The Elements of Statistical Learning*. Springer.

M. Hu and B. Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*, pages 168–177, Seattle, WA, USA.

K. Järvelin and J. Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446.

J. J. Jiang and D. W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of ROCLING*, pages 19–33, Taiwan, China.

Y. Jo and A. H. Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of WSDM*, pages 815–824, Hong Kong, China.

R. M. Karampatsis, J. Pavlopoulos, and P. Malakasiotis. 2014. AUEB: Two stage sentiment analysis of social network messages. In *Proceedings of SemEval*, Dublin, Ireland.

R.M. Karampatsis. 2012. Name entity recognition in Greek texts from social networks.

S.-M. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of COLING*, pages 1367–1373, Geneva, Switzerland.

S.-M. Kim and E. Hovy. 2006. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of SST*, pages 1–8, Sydney, Australia.

I. P. Klapaftis and S. Manandhar. 2010. Taxonomy learning using word sense induction. In *Proceedings of NAACL*, pages 82–90, Los Angeles, CA, USA.

N. Kobayashi, K. Inui, and Y. Matsumoto. 2007. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proceedings of EMNLP-CoNLL*, pages 1065–1074, Prague, Czech Republic.

D. Lin and X. Wu. 2009. Phrase clustering for discriminative learning. In *Proceedings of ACL*, pages 1030–1038, Suntec, Singapore. ACL.

D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of ICML*, pages 296–304, Madison, WI, USA.

B. Liu, M. Hu, and J. Cheng. 2005. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of WWW*, pages 342–351, Chiba, Japan.

B. Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool.

C. Long, J. Zhang, and X. Zhut. 2010. A review selection approach for accurate feature rating estimation. In *Proceedings of COLING*, pages 766–774, Beijing, China.

P. Malakasiotis, R. M. Karampatsis, K. Makrynioti, and J. Pavlopoulos. 2013. nlp.cs.aueb.gr: Two stage sentiment analysis. In *Proceedings of SemEval*, pages 562–567, Atlanta, Georgia, U.S.A.

C. D. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. 1993. Building a large annotated corpus of English: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Q. Mei, X. Shen, and C. Zhai. 2007. Automatic labeling of multinomial topic models. In *Proceedings of KDD*, pages 490–499, San Jose, CA, USA.

T. Mikolov, C. Kai, G. Corrado, and J. Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*.

T. Mikolov, W.-T. Yih, and G. Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL HLT*.

G.W. Milligan. 1980. An examination of the effect of six types of error perturbation on fifteen clustering algorithms. *Psychometrika*, 45(3):325–342.

S. Moghaddam and M. Ester. 2010. Opinion digger: an unsupervised opinion miner from unstructured product reviews. In *Proceedings of CIKM*, pages 1825–1828, Toronto, ON, Canada.

S. Moghaddam and M. Ester. 2012. On the design of LDA models for aspect-based opinion mining. In *Proceedings of CIKM*, pages 803–812, Maui, HI, USA.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. 29(3):436–465.

S. M. Mohammad, S. Kiritchenko, and X. Zhu. 2013. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of SemEval*, Atlanta, Georgia, USA.

R. Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1–10:69.

F. A. Nielsen. 2011. A new anew: evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of ESWC*, pages 93–98, Heraclion, Greece.

O. Owoputi, B. OŠConnor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL*, Atlanta, Georgia.

S. Padó and M. Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

B. Pang and L. Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*, Barcelona, Spain.

B. Pang and L. Lee. 2005. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124, Ann Arbor, MI, USA.

B. Pang and L. Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

J. Pavlopoulos and I. Androutsopoulos. 2014. Aspect term extraction for sentiment analysis: New datasets, new evaluation measures and an improved unsupervised method. In *Proceedings of the EACL Workshop on Language Analysis for Social Media*, pages 44–52, Gothenburg, Sweden.

T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. Wordnet::similarity: measuring the relatedness of concepts. In *Proceedings of NAACL:HTL*, pages 38–41, Boston, MA, USA.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manadhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of SemEval*, Dublin, Ireland.

A.-M. Popescu and O. Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of HLT-EMNLP*, pages 339–346, Vancouver, Canada.

J. R. Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1(1):81–106.

P. Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65.

T. Sakai. 2004. Ranking the NTCIR systems based on multigrade relevance. In *Proceedings of AIRS*, pages 251–262, Beijing, China.

B. Snyder and R. Barzilay. 2007. Multiple aspect ranking using the good grief algorithm. In *Proceedings of NAACL*, pages 300–307, Rochester, NY, USA.

R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y Ng, and C. Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642, Seattle, WA, USA.

I. Titov and R. T. McDonald. 2008a. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-HLT*, pages 308–316, Columbus, OH, USA.

I. Titov and R. T. McDonald. 2008b. Modeling online reviews with multi-grain topic models. In *Proceedings of WWW*, pages 111–120, Beijing, China.

M. Tsytsarau and T. Palpanas. 2012. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery*, 24(3):478–514.

P. D. Turney. 2002. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL*, pages 417–424, Philadelphia, PA.

V. Vapnik. 1998. *Statistical Learning Theory*. Wiley.

C.-P. Wei, Y.-M. Chen, C.-S. Yang, and C. C Yang. 2010. Understanding what concerns consumers: a semantic approach to product feature extraction from consumer reviews. *Information Systems and E-Business Management*, 8(2):149–167.

T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT-EMNLP*, pages 347–354, Vancouver, BC, Canada.

Z. Wu and M. Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of ACL*, pages 133–138, Las Cruces, NM, USA.

J. Yu, Z. Zha, M. Wang, and T. Chua. 2011a. Aspect ranking: identifying important product aspects from online consumer reviews. In *Proceedings of NAACL*, pages 1496–1505, Portland, OR, USA.

J. Yu, Z. Zha, M. Wang, K. Wang, and T. Chua. 2011b. Domain-assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews. In *Proceedings of EMNLP*, pages 140–150, Edinburgh, UK.

T. Zesch and I. Gurevych. 2010. Wisdom of crowds versus wisdom of linguists - measuring the semantic relatedness of words. *Natural Language Engineering*, 16(1):25–59.

Z. Zhai, B. Liu, H. Xu, and P. Jia. 2010. Grouping product features using semi-supervised learning with soft-constraints. In *Proceedings of COLING*, pages 1272–1280, Beijing, China.

Z. Zhai, B. Liu, H. Xu, and P. Jia. 2011. Clustering product features for opinion mining. In *Proceedings of WSDM*, pages 347–354, Hong Kong, China.

Z. Zhang, A. Gentile, and F. Ciravegna. 2013. Recent advances in methods of lexical semantic relatedness – a survey. *Natural Language Engineering*, FirstView(1):1–69.

W. X. Zhao, J. Jiang, H. Yan, and X. Li. 2010. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proceedings of EMNLP*, pages 56–65, Cambridge, MA, USA.