



UNIVERSITEIT VAN AMSTERDAM

MASTERS THESIS ARTIFICIAL INTELLIGENCE

A Semi-Supervised Approach to Context-Based Sentiment Analysis

Author:

Hossein Kazemi

Supervisors:

dr. Maarten van Someren - **Universiteit van Amsterdam**

dhr. Matthijs Mulder - **ParaBotS B.V.**

Submission Date:

July 31, 2011

Abstract

This paper mainly discusses a new approach to solve the Feature-Based Sentiment Analysis problem by combining powerful methods from Topic-Modeling, Ontologies and Active Learning. Our approach decomposes the sentiment analysis task into two subtasks. The first task is to recognize the concept to which a sentence or a phrase is referring to using a Topic Modelling approach. The second task is to classify the sentiments of a fragment of text (in our case sentences) towards each of these recognized concepts. First, the “bag-of-words” feature representation is made from each of these sentences as the feature vectors. Next, extra dimensions are added to these feature vectors with each dimension corresponding to a particular concept. Then these newly made feature vectors are given to our Active Learning algorithm as inputs. Our Active Learning algorithm trains a classifier to classify sentiments, and since the concepts are included in the feature vectors, it classifies sentiments based on their concepts (in other words, based on the “context” in which the sentiment words are being used.). We observed that, in general, a combination of context, sentence words and sentiment is needed for a correct sentiment classification. If we do not include concepts, then a word in a sentence will be treated in the same way for all classes (positive or negative). However, in this way we classify a pair <phrase, concept> as positive or negative instead of only classifying a phrase as having positive or negative sentiment. Our classification results show that our method significantly outperforms the base-line.

In the end a summary of the opinions is generated using our domain ontology, the recognized concepts and the sentiment labels which were determined by our sentiment classifier as the final output of the system.

Contents

1	Introduction	4
1.1	Concepts and Problem Definition	6
2	Previous work	8
2.1	Previous Work on general Sentiment Analysis	8
2.2	Previous work on Feature-based Sentiment Analysis	10
2.2.1	General Methods	10
2.2.2	Ontology-Based methods	11
2.2.3	Topic-Modeling based methods	13
2.3	Motivation and research questions	15
3	Approach	17
3.1	Concept detection	17
3.1.1	Domain Ontology	17
3.1.2	Ontology Population	23
3.1.3	Concept recognition	23
3.2	Sentiment Classification	23
3.2.1	Construction of Feature Vectors	23
3.2.2	Active Learning	24
3.3	Opinion Summarization	26
4	Experimental Setup	28
4.1	The restaurant review domain dataset	28
4.2	Preprocessing	29
4.3	Training, Development and Test sets	29
4.4	Concept detection	31
4.5	Sentiment Classification	31
4.6	Supervised method(base-line)	32
5	Results and Analysis:	33
5.1	Concept detection results and analysis	33
5.2	Sentiment Classification phase results and analysis	37
5.2.1	Results for different decision boundaries	37
5.2.2	Results in terms of number of labeled datapoints	39
5.2.3	Results in terms of inclusion of concepts into feature vectors	39
5.3	Opinion Summarization	42
6	Future work	44

7 Conclusion	44
A Appendix	46
A.1 Seed Lists	46
A.2 Table translations to English	46

List of Figures

1 Domain Ontology for restaurant reviews	18
2 Feature vector construction diagram	25
3 The workflow of our system for producing opinion summaries as final outputs	28
4 A partial example of an opinion summary	43

List of Tables

1 Corpus Statistics	29
2 Corpus sample statistics: number of times that each concept is used in the sample and the number of positive and negative sentiments	30
3 A short list of seed words used for each concept	31
4 Top words in each topic	34
5 Results of modified LDA on the test set	35
6 A few examples for which the concepts where correctly rec- ognized	36
7 A few examples for which the concepts where incorrectly rec- ognized	36
8 Results for Sentiment Classification	38
9 Comparison of results with base-line	39
10 Classification comparison examples(Pos:Positive,Neg:Negative)	40
11 English translation of Table 5	46
12 English translation of Table 6	47
13 English translations of Table 9	47

1 Introduction

Today, there is a large amount of available data on the Internet and there is an ever increasing demand to mine these data in order to elicit useful information out of them. Sentiment Analysis or Opinion Mining is one of the recently most popular tasks in this realm. Extracting and mining people's opinions about a person or a certain product, mining people's reactions to the events and news etc. fall into the category of Sentiment Analysis.

Evidently, the web has dramatically changed the way people express their views and opinions. People can now post reviews of products and express their views on almost anything on the Internet forums, discussion groups, blogs and tweets, which are collectively called the *user-generated contents*. The impact is so strong that nowadays if one wants to buy a certain product or go to a restaurant, he is no longer limited to asking his friends or family because he can easily have access to a very large number of product or restaurant reviews on the Web [40].

At the same time, major companies or even small retailers are increasingly coming to realize that consumer voices in the form of online opinions can have enormous impacts on improving the quality of their services or productions to better compete in the market. It is also worth noting that services or consumption of goods is not the only motivation behind people's seeking out or expressing opinions online. A need for political information is also another important factor.

However, finding opinion sources and monitoring them on the web can still be a formidable task because there is a large number of diverse sources, and each source may also have a huge volume of *opinionated text* which is being generated in seconds. In many cases, opinions are hidden in long forum or blog posts, and therefore, it is difficult for a human reader to find relevant sources, extract related materials with opinions, read them, summarize them and organize them into presentable and usable forms [40]. This is the main driving motivation for developing automated opinion discovery and summarization systems. In other words, the need for Sentiment Analysis.

One of the main challenges in Sentiment Analysis is that sentiment words¹

¹Words that signal positivity or negativity. They will be discussed in the next chapter in detail

may signal different sentiment polarities² when they are used in different “contexts” in the text. In the case of Feature-Based Sentiment Analysis³, sentiment words may signal different polarities when they are used on different features. For example, in the domain of restaurant reviews, the sentiment polarity of the word “*koud*” (cold) when it is used for the feature “*bier*” (beer) in the phrase “*koud bier*” (cold beer) is positive and when it is used for the feature “*personeel*” (personnel) in the phrase “*koud personeel*” (cold personel) is negative. Tackling this problem requires additional measures and maybe new approaches in the area of Feature-Based Sentiment Analysis.

This paper mainly discusses a new approach to solve the Feature-Based Sentiment Analysis problem by combining powerful methods from Topic-Modeling, Ontologies and Active Learning. This approach decomposes the sentiment analysis task into two subtasks. The first task is to recognize the concept to which a sentence or a phrase is referring to. In the case of Feature-Based Sentiment Analysis, these concepts can be referred to as *features*⁴. Actually, the first task involves recognizing these features. The second task is to classify the sentiments of a fragment of text (in our case sentences) towards each of these recognized concepts. First, the “bag-of-words”⁵ feature representation is made from each of these sentences as the feature vectors. Next, extra dimensions are added to these feature vectors with each dimension corresponding to a particular concept. Then these newly made feature vectors are given to our Active Learning algorithm as inputs. Our Active Learning algorithm trains a classifier to classify sentiments, and since the concepts are included in the feature vectors, it classifies sentiments based on their concepts (in other words, based on the “context” in which the sentiment words are being used.). This algorithm trains a classifier in iterations, starting from a small set of labeled datapoints as seeds. In the end a summary of the opinions is generated using our domain ontology, the recognized concepts and the sentiment labels which were determined by our sentiment classifier. Our work is based on the restaurant reviews (in Dutch language)

²positivity or negativity

³Feature-Based Sentiment Analysis is a branch of work in Sentiment Analysis which focus on determining sentiment polarities on features. For definition and details refer to 1.1 and 2.2.

⁴Although in general the words *concept* and *feature* have different meanings, in this paper they refer to the same thing and therefore, they are interchangeably used throughout the text

⁵In this representation, a text (such as a sentence or a document) is represented as an unordered collection of words, disregarding grammar and even word order.

obtained from `www.iens.nl`.

Our main idea is that by directly including the concepts in the learning process we can distinguish between the polarity of certain sentiment words that may signal different polarities when used in different “context”. Following the previous example, distinguishing between the sentiment polarity of the word *koud* in *koud bier* and *koud personeel*.

Although there have been attempts like [31] to build topic-specific sentiment lexicons to be used for context based sentiment classification, they weren’t much successful. Moreover, state-of-the-art Sentiment Analysis systems that use machine learning approaches, as we will discuss in the next chapter, are not able to make this distinction to correctly classify sentiments when there is a change in the context in the same corpus being studied. We believe that we can improve the sentiment classification results with our approach. Moreover, our method is able to do sentiment classification on implicit features (in addition to explicit features)⁶ which also helps in generating a more cohesive opinion summary. Previous work focuses on explicit features and mostly uses Natural Language Processing techniques for this. To the best of our knowledge, this work is the first attempt to do Sentiment Analysis in this manner. We compared our results with a fully supervised method as our baseline and the results show that our method significantly outperforms it.

1.1 Concepts and Problem Definition

In this section, a brief description of the concepts used in this field and throughout this paper will be given and based on them, a formal description of the problem is provided. Our terminology and definitions are similar to [39].

Definition: object and features In general, opinions can be expressed on any entity such as a product, an individual, an organization, an

⁶Explicit features are features that are directly mentioned in a phrase, and implicit features are features that there is no explicit indication of it in the phrase. Refer to 1.1 for more details and examples.

event or a specific topic. This entity is called an *object*. Each object can have one or more *features*⁷. In general, each feature can be *implicitly* or *explicitly* mentioned in the opinionated text fragment being studied. For example, in the case of opinions for a restaurant, “*bediening*” (service) in “*prima bediening*” (top service) is an *explicit* feature because the feature is directly mentioned in the phrase. On the contrary, the phrase “*vriendelijk en snel*” (friendly and quick) also talks about the same feature as the previous example “*bediening*” without any explicit indication of the feature. This is a very common example of an *implicit* feature.

Definition: opinion and opinion orientation An *opinion* on an object or a feature is the view, attitude, emotion or appraisal towards the object or feature. And the *opinion orientation*⁸ indicates the positivity, negativity or neutrality of the opinion towards the object or its feature(s).

Problem definition: Generally the task of Sentiment Analysis boils down to classifying a fragment of text that has an opinion towards an object or its features, as positive or negative. This is called *Sentiment Classification*⁹.

On the one hand, this classification can be done on a document basis in which case sentiment orientation of the document (which is about a certain object or feature) is decided based on the overall sentiments of the sentences in the document. On the other hand, this classification can be based on the sentiment orientation of individual sentences. In the literature, the former is called *Document-level Sentiment Classification* and the later *Sentence-level Sentiment Classification*.

The work presented in this paper goes beyond the two aforementioned levels and does sentiment classification based on phrases also in addition to sentences.

⁷More precisely, an object is associated with a set of components (or parts) and their attributes. An opinion can be expressed on both components and their corresponding attributes. However, the term *feature(s)* is used for both in the literature. Moreover, features are also called *aspects* in certain papers

⁸opinion orientation, sentiment orientation and semantic orientation are interchangeably used in the literature [40].

⁹Another major sub-task in sentiment analysis is *subjectivity classification* which is classifying a text fragment (usually a sentence) as being neutral or opinionated. However, a detailed discussion of it is out of the scope of this paper.

The next chapter succinctly describes the state-of-the-art and some of the most influential works in the field of sentiment analysis and opinion mining.

2 Previous work

Because of the popularity of the field there has been a plethora of work done, each trying to solve the sentiment analysis problem from a different perspective and a different method to fulfil a certain goal. Below the description of the most relevant works to this project is given:

2.1 Previous Work on general Sentiment Analysis

As said before, the general Sentiment Analysis problem boils down to *Sentiment Classification*. Works such as [44][6][17][19][21][24][26] all focus on sentiment classifications.

Perhaps [44] is the most cited work on document-level sentiment analysis. They consider the problem of classifying movie reviews by their overall sentiments using several supervised machine learning techniques (Naive Bayes, Maximum Entropy classification and Support Vector Machines) and reporting an accuracy of 82.7%. They use documents as data instances and represent the texts in these documents as feature vectors where dimensions are presence or absence of words (bag-of-words). Each document is labeled as having a positive or a negative sentiment. In similar attempts, adjectives have been employed as features by a number of researchers such as [43][55]. The difference between these methods and the previous one is that they only use adjectives to form their feature vectors. In other words, each document is represented as a feature vector containing only the adjectives as features. While [44]’s method uses all words (nouns, adjectives, adverbs, verbs, etc.) to form the feature vectors. Rather than focusing on isolated adjectives, [53] proposes detecting the document sentiments based on selected phrases, where the phrases are chosen via a number of pre-specified part-of-speech patterns, most of them containing an adjective or an adverb.

In all supervised approaches, reasonably high accuracy can be obtained only subject to the requirement that test data be similar to training data. To

move a sentiment classifier to another domain would require collecting annotated data in the new domain and retraining the classifier. This dependency on annotated training data is one of the major shortcoming of all supervised methods. The reason is that words and even language constructs used in different domains for expressing opinions can be substantially different. In addition the same word in one domain may have a positive meaning, but in another domain may have a negative meaning. This is the same problem with “*koud personeel*” and “*koud bier*” (as we mentioned before) that we are trying to tackle.

Non-machine learning unsupervised approaches to sentiment classification can solve the problem of domain dependency and reduce the need for annotated training data. The most notable work is [53] which uses two arbitrary seed words (poor and excellent) to calculate the semantic orientation of phrases, where the orientation of a phrase is defined as the difference of its association with each of the seed words (as measured by point-wise mutual information (PMI)¹⁰). The sentiment of a document is calculated as the average semantic orientation of all such phrases. [56] describes a method of automatic seed word selection for unsupervised sentiment classification of product reviews in Chinese. The method only requires information about commonly occurring negations and adverbials in order to iteratively find sentiment bearing items.

The problem with the above methods -both at document-level and sentence-level- is that they do not provide the necessary detail needed for some applications. In other words, a positive or negative opinionated document or sentence on a particular object does not mean the author has a positive or negative opinion about all features of the object. Typically, the author of the opinion might use mixed sentiments towards different features of an object, although the general sentiment on the object may be positive or negative [40].

Another important issue with the above methods at document-level is that they all assume that the entire text of a document expresses an opinion only about a certain object or a feature. However, there are also texts that are about many things and a subtask is to find out which part of the text is about the target object or feature.

¹⁰PMI is a measure of association between two words in a corpus

To fulfill the need to solve the aforementioned problems, another set of research has gone to the objects’ feature level and is called “Feature-based Sentiment Analysis”. Some of the most notable previous work on this is discussed in the next section.

2.2 Previous work on Feature-based Sentiment Analysis

2.2.1 General Methods

Based on the definitions in chapter 1, feature-based sentiment analysis mainly involves identifying *object features* that have been commented on and then determining whether the opinions on the features are positive, negative or neutral. For instance, in the sentence “The food at this restaurant is awesome”, the object feature is “food” and the opinion on it is “positive”.

Usually solving the former is formulated as a *Named Entity Recognition* problem in the information extraction community. In general information extraction, there are two main approaches: rule-based and statistical. Early extraction systems are mainly based on rules. In statistical methods, the most popular models are Hidden Markov Models (HMM) [48], Maximum Entropy Models (ME) [14] and Conditional Random Fields (CRF) [36]. By far, CRF has been shown to be the most effective method. However, they all require a large amount of labeled data to produce acceptable results and even CRF does not perform well when it comes to feature and opinion word pairs that have long range dependencies [47].

[30] proposed a technique based on association rule mining to extract product features. Their main idea is that people often use the same words when they comment on the same product features. Then frequent itemsets of nouns in reviews are likely to be product features while the infrequent ones are less likely to be product features. Then they utilize the adjective synonym set and antonym set in WordNet [42] to predict the semantic orientations of adjectives. Although their results look promising, they heavily rely on WordNet and although they criticize other works for using large corpus, they themselves rely on a relatively large amount of text.

Following [30], several researchers have further explored the idea of using opinion words in product feature mining. A dependency based method is

proposed in [59] for a movie review analysis application. However, they rely on manually built rules and dependency relation templates.

In [46] an unsupervised information extraction system which mines reviews in order to build a model of important product features is introduced. The authors use *relaxation labeling* for finding the semantic orientation of words in the text. They report relatively good results. However, the feature extraction part requires huge amounts of computations and their system needs to constantly query the web which is extremely time-consuming. Also, the need for manually writing rules to help the feature extraction process can be counted as another drawback. More importantly, although they claim that their sentiment classification method (*relaxation labelling*) is unsupervised, they provide labels for training and heavily rely on manually-supplied *syntactic dependency rule templates* and WordNet[42].

The main limitation of these approaches is that there are many extracted features and there is a lack of organization. Thus, similar features are not grouped together (for example, in a restaurant domain, “*sfeer*” (atmosphere) and “*ambience*” (ambiance)) and possible relationships between features of an object are not recognized (for example, “*koffie*” is a kind of “*drank*” (drink)). In addition polarity analysis (positive,negative or neutral) of the text fragment being examined is done by assigning the dominant polarity of opinion words it contains (usually adjectives), regardless of *polarities individually associated to each feature*.

The lack of organization as mentioned before, was the main driving motivation for another set of work which is introduced next.

2.2.2 Ontology-Based methods

Due to the aforementioned shortcomings of previously mentioned methods, some researchers focused on using a taxonomy or an ontology in their systems. One of the initial works in this area is [12]. In their method, they use predefined taxonomies and semantic similarity measures to automatically extract known features of a product and calculate how close to predefined concepts in the taxonomy they are (using WordNet). However, this should be reviewed by a user in order to insert missing concepts in the right place while avoiding duplications which requires a lot of care and is not really

practical. Moreover, the steps of identifying opinions and their polarity and the production of a summary are not detailed.

Other notable works that use feature taxonomies are [25] and [8]. Their results show that the use of a taxonomy significantly improves the quality of extracted features. However, for sentiment classification they rely on a high number of manually labeled data in addition to using the WordNet lexicon.

As mentioned previously, in addition to taxonomy-based methods, several researchers have focused on ontology-based techniques. The feature extraction phase is guided by a domain ontology, built either manually [57] or semi-automatically [13], which is then enriched by an automatic process of extraction/clustering of terms which correspond to new feature identification.

The system OMINE in [13] proposes a mechanism for ontology enrichment using a domain glossary which includes specific terms such as words of jargon, abbreviations and acronyms. This ontology enrichment step is called *ontology lexicalization* in the literature. They use manually written rules to *lexicalize* their “offline” built ontology. In addition to using their ontology to extract features, they heavily rely on an external information extraction engine called *SProUT* [7] which they have adapted for their needs to improve the feature extraction results. They also use a sentiment lexicon to do sentiment classification on features. Their results look promising which they attribute to their use of the aforementioned information extraction engine.

In [57] the authors use a corpus based method to add to their ontology concepts: sentences containing a combination of conjunction words¹¹ and already recognized concepts are extracted. This process is repeated iteratively until no new concepts are found. In their method they use SentiWordNet¹² [23], and successfully apply it to the IMDB movie review corpus. The major drawback of their work is that they also rely on an external resource (SentiWordNet) which limits the range of work only to English language, as there is no SentiWordNet published for other languages (such as Dutch) yet.

Probably, the most recent ontology-based sentiment analysis system is in-

¹¹Conjunction words are **and**, **but**, **or**,...

¹²SentiWordNet is a lexical resource for opinion mining. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity

troduced in [11]. They use a domain ontology to produce comprehensive summaries of opinions for restaurant reviews. Their approach is mainly based on the previous work in [5]. Their idea is that a review R is composed of a set of elementary discourse units (EDU). An EDU is a clause containing one elementary opinion unit (EOU) or a sequence of clauses that together bear a rhetorical relation to a segment expressing an opinion. They segment each review in EDUs using their discourse parser mentioned in [1]. Then for each EDU, the system extracts EOUs using a rule based approach. Next, it extracts features that correspond to the process of term extraction using domain ontology. After that, for each feature within an EDU, it associates the set of opinion expressions and finally produces a discourse-based summary. However, they do not provide any results for the summarization in the end. Moreover, they heavily rely on a manually labeled corpus and manually built extraction rules for extracting elementary opinion units.

The main drawback to ontology-based sentiment analysis methods is that, overall, extracted features correspond exclusively to terms contained in the ontology. In other words, the more populated (accurately) the ontology the more features to be extracted in the end. However, studies show that the use of an ontology/taxonomy significantly improves the results compared to general feature-based sentiment analysis methods. Moreover, so far, the previous work on ontology-based sentiment analysis has been mainly focused on manually written rules or external tools such as a domain specific lexicon, WordNet or manually built gazetteers to populate/lexicalize the ontology. There has been a lot of work done for populating the ontologies that still haven't been used in combination with sentiment analysis approaches. Some of these will be succinctly discussed in the next chapter as discussing them in this chapter will deviate the reader from the main topic.

2.2.3 Topic-Modeling based methods

The final range of work which has been much in attention in recent years is the use of *Topic Modeling* approaches in sentiment analysis. They mainly focus on *aspect*¹³ *discovery* and *domain adaptation of sentiment words*.

¹³a multinomial distribution over words that represents a more specific topic in reviews, this is actually called a feature as defined in the previous section. Words *aspect*, *feature*, *topic* and *concept* refer to the same concept in this paper and they are interchangeably used throughout the text.

Works such as [41] and [38] are among the most notable ones. [41] proposes a Topic-Sentiment Mixture (TSM) model that can reveal the latent topical facets in a weblog collection, the sub-topics in the results of an ad-hoc query, and their associated sentiments. However, their method requires a lot of labelled (positive, negative) data for each topic. Moreover, TSM is essentially a form of probabilistic Latent Semantic Analysis (PLSA) [29], thus suffers from problems of inference on new document and overfitting the data [38] (this will be discussed more in detail in the following section).

Along this line, [38]’s work is based on *Latent Dirichlet Allocation(LDA)* [9]. In their work, they propose a joint sentiment and topic model, called Sentiment-LDA, by adding a sentiment layer to the normal LDA (see the paragraph for Latent Dirichlet Allocation in section 3.1.1 for details). They observe that sentiments are dependent on local context. Therefore, to capture this dependency in the document, they propose Dependency-Sentiment-LDA by considering the inter-dependency of sentiments through a Markov chain. They use 5 sentiment lexicons in combination as prior knowledge for their method. However, the topics that are produced are a combination of features and sentiment words, and they do not provide any details on how they perform the sentiment classification for which the results are relatively low.

Probably the latest work exploiting topic-modelling techniques is [32]. Their proposed method *Aspect and Sentiment Unification Model(ASUM)* incorporates aspect and sentiment together to model sentiments towards different aspects which actually tries to discover pairs of $\{aspect, sentiment\}$ (called senti-aspects). They use a supervised method for Sentiment classification, and their results are impressive.

In general, the major disadvantage of employing topic-modelling approaches is that they mainly focus on rediscovering topics or sentiment words from a corpus. Their output is usually a list of most probable words for each topic and each topic is independent of the other topics. The problem is that they do not have the necessary mechanism to hold the possible structure between the topics. It would be very beneficial, especially when there is a need for opinion summarization and grouping of features to somehow have a structure (mostly in terms of taxonomies) between these independent topics.

2.3 Motivation and research questions

So far, a very brief survey of previous work relative to this paper is given. The main drawbacks of these methods can be categorized as follows:

1. On the one hand, supervised methods require a lot of labeled data which is not available most of the time and is expensive in time and cost to obtain.
2. On the other hand, unsupervised methods usually require a large amount of data and processing resources and are heavily relying on external tools such as SentiWordNet, manually built lexicons or information extraction engines. Which might not be easily available to the public, or in other languages such as Dutch, especially for commercial uses when it comes to companies.
3. Moreover, in ontology-based sentiment analysis methods, overall, extracted features correspond exclusively to terms contained in the ontology. And since previous works on ontology-based sentiment analysis has been mainly focusing on manually written rules or external tools such as domain specific lexicons, WordNet or manually built gazetteers to populate/lexicalize the ontology, a need is being felt to incorporate stronger unsupervised ontology population/lexicalization methods into these kind of approaches for sentiment analysis.
4. Also, the lack of a proper organization and grouping of concepts and sentiments discovered by topic-modelling approaches, makes the need for a proper organization mechanism more evident.
5. To the best of our knowledge, almost all of the previous methods work on explicit features and are not able to recognize implicit features¹⁴ and their sentiments. Furthermore, with the previous approaches it is almost impossible to do sentiment analysis on certain sentences. For example, in *“wij moeten erg lang op onze drankjes wachten”* (we had to wait a long time for our drinks) the concept of this sentence actually refers to the *“service”* of a restaurant and not drinks. Therefore, the negative sentiment is towards *“service”* respectively. Another example would be *“we werden hartelijk ontvangen”* (we were warmly welcomed) which is again impossible for the aforementioned approaches to classify.

¹⁴Implicit and explicit features were discussed in chapter 1.

6. As far as we know, previous methods can not classify sentiments based on the “context” in which the sentiment words are being used. In other words, they are not able to correctly classify sentiments when there is a change in the context in the same corpus being studied. For example, they cannot distinguish between the sentiment polarity of the word “*koud*” in “*koud bier*” and “*koud personeel*”. As in the first phrase “*koud*” signals a positive sentiment and in the second case a negative sentiment.

Having said above, this paper attempts to answer these research questions:

1. Is it possible to detect implicit features of an object in reviews ?
2. Does directly including the concept (to which a sentence refers) in the learning process allow the learner to classify sentiments based on their context (concept) in which they are being used?

Our approach attempts to answer the questions above without using external tools or resources, and using as few labeled data as possible. We mainly build upon the state-of-the-art machine learning techniques in the sentiment analysis problem, to increase the accuracy of the sentiment classification. We discuss it in the next chapter.

3 Approach

In this section we give a more detailed description of our approach and we introduce some of the concepts and methods involved.

Our approach tries to look at the Sentiment Analysis problem in a different way. Basically, we decompose the Sentiment Analysis task into two phases. First, our approach tries to recognize the concept (from a domain ontology about restaurants) to which a sentence or phrase is mostly related (Concept detection phase), and then by directly including the concepts in the feature vectors, we attempt to do sentiment classification using an Active Learning methodology (Sentiment Classification phase). This two step approach gives us the ability of classifying sentiments based on their “context” in which they are being used. In the end, we use our domain ontology to produce a summary of opinions. Our work is based on the restaurant reviews obtained from www.iens.nl as mentioned before.

3.1 Concept detection

Our concept detection phase involves three main components:

1. A domain Ontology
2. A module to populate/lexicalize this domain Ontology
3. A Concept recognition module

3.1.1 Domain Ontology

In theory, an ontology is a “formal, explicit specification of a shared conceptualisation” [28]. In other words, an ontology is a formal representation of knowledge as a set of concepts within a domain, and the relationships between those concepts. In the case of “restaurants” as the domain, concepts usually cover the gamut from staff and interiors to drinks and food, depending on the granularity. However, when the domain comes to “restaurant reviews” then the concepts involved are usually limited to the concepts that “reviewers” mostly talk about. Our studies on the obtained reviews show that the most used concepts in this domain are: “*Drank, Voorgerecht, Hoofdgerecht, Dessert, Sfeerkenmerken*”¹⁵ and *Bediening*”. For this reason, our domain Ontology for restaurant reviews was made based on these concepts. Figure 1 shows the hierarchy of our ontology.

¹⁵ *Voorgerecht*: appetizer, *Hoofdgerecht*: main dish, *Sfeerkenmerken*: atmosphere

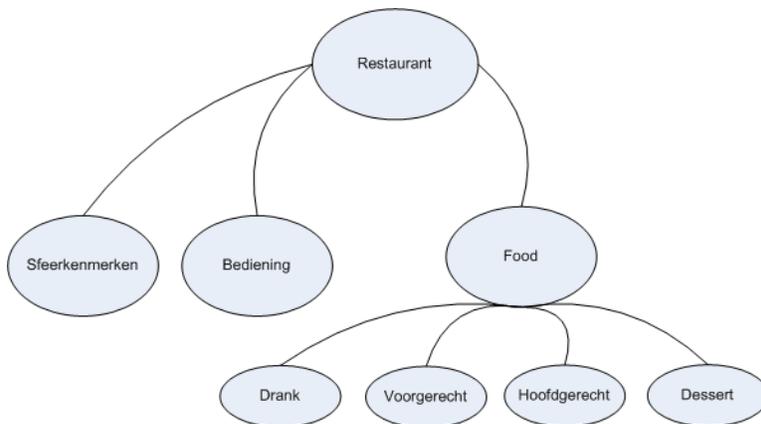


Figure 1: Domain Ontology for restaurant reviews

The hierarchy between the concepts in our ontology is used to produce summarized opinions towards each of these concepts which we discuss later. To achieve this we need to populate/lexicalize our Ontology. Ontology population is the process of inserting concept instances into an existing ontology. In a simplified view, an ontology can be thought of as a set of concepts, relations among the concepts and their instances. A concept instance is a realization of the concept in the domain, e.g. the instantiation of the concept as a phrase in a textual corpus. The process of ontology population does not change the structure of an ontology, i.e., the concept hierarchy and non-taxonomic relations are not modified. What changes is the set of realizations (instances) of concepts in the domain.

There has been a lot of research on ontology population which can be found in [15] for an extensive survey. Recently, probabilistic Topic Models received much attention [45][54]. They have been primarily used in document modeling, topic extraction, and classification purposes in Information Retrieval. The most important advantages of these methods are that they are completely unsupervised and do not rely on any external tools or resources which are quite suitable for the purpose of this project.

For the sake of clarity, we first give a brief introduction to Topic Models, gradually leading our discussion to our ontology population and concept recognition modules. We think that introducing Topic Models in this section is more suitable for the readers to better understand the motivation behind

choosing these methods in our work.

Topic Models In machine learning and natural language processing, a topic model is a type of statistical model for discovering the abstract “topics” that occur in a collection of documents. This field started with the introduction of Latent Semantic Analysis (LSA) [20] to overcome the shortcomings of conventional Information Retrieval techniques with regard to computation of similarity between documents. The main idea behind LSA is to transform the document representation from a high-dimension word space to a low-dimension word latent semantic space in order to capture implicit structures in the association of terms with documents. However, LSA has an inherent weakness in that although the words and documents can be represented as points in euclidean space, its results do not introduce well defined probabilities and thus, are difficult to interpret [29][54].

Variants of the LSA such as probabilistic Latent Semantic Analysis (pLSA) [29] and Latent Dirichlet Allocation(LDA) [9] have been developed to improve the interpretation of the results generated by LSA. In the statistical models of topics, semantic properties of words and documents are represented using probabilistic (latent) topics and the internal structure of the text is interpreted using word-topic and topic-document distributions.

The probabilistic Latent Semantic Analysis(pLSA) is a generative statistical model for analyzing co-occurrence of data which associates a latent variable z with an observation (i.e., each occurrence of the word w in document d). The generative process of the pLSA is defined as:

1. Choose a document d with a prior probability $p(d)$
2. Choose a latent class z from the document with probability $p(z|d)$
3. Choose a word w from the latent class distribution with probability $p(w|z)$

Parameter estimation in the pLSA is based on the likelihood principle and is approximated using Expectation-Maximization(EM) algorithm.

Discussion: While the pLSA represents a significant step towards probabilistic modeling of text, it is incomplete in that it provides no probabilistic model at the level of documents [9]. In other words, the model does not have any control over how mixture weights $p(z|d)$ are generated. The limitation leads to two problems: the number of parameters that need to be estimated grows linearly with the size of the corpus, which leads to overfitting [9][54][34]. In addition, although PLSA is a generative model of the documents in the collection it is estimated on, it is not a generative model of new documents; therefore, it is difficult to test generalizability of the model to new documents [51].

Latent Dirichlet Allocation The Latent Dirichlet Allocation [9] is an attempt to improve the pLSA by introducing a Dirichlet prior on document-topic distribution. As a conjugate prior¹⁶ for multinomial distributions [16], Dirichlet prior simplifies the problem of statistical inference. [27] [51] explore a variant of the LDA by placing a symmetric Dirichlet prior on the topic-word distribution, and demonstrate how to perform parameter estimation using Gibbs sampling, a form of the Markov Chain Monte Carlo [2] (will be discussed next).

The generative process in the LDA is similar to pLSA: each word w in a document d is generated by sampling a topic z from topic distribution, and then sampling a word from the topic-word distribution.

Let there be T topics and $\mathbf{w} = w_1 \dots w_n$ represent a corpus of D documents, with a total of n words. We use d_i to denote the document of word w_i , and z_i for the hidden topic from which w_i is generated. Then we can have:

$$P(w_i) = \sum_{j=1}^T p(w_i|z_i = j)p(z_i = j), \quad (1)$$

where $p(z_i = j)$ is the probability that j th topic is sampled for the i th word token¹⁷, and $p(w_i|z_i = j)$ is the probability of sampling w_i under topic j . Intuitively, $p(w|z)$ indicates which words are important to a topic, whereas $P(z)$ is the prevalence of those topics within a document.

¹⁶In Bayesian probability theory, if the posterior distributions $p(\theta|x)$ are in the same family as the prior probability distribution $p(\theta)$, the prior and posterior are then called conjugate distributions, and the prior is called a conjugate prior for the likelihood.

¹⁷A word token is an occurrence of an indexed word in the training corpus

Let $\phi_j^{(w)} = p(w|z = j)$, and $\theta_j^{(d)} = p(z = j)$ for document d . We can formally define the LDA generative model as below:

$$\theta \sim \text{Dirichlet}(\alpha) \quad (2)$$

$$z_i | \theta^{(d_i)} \sim \text{Multinomial}(\theta^{(d_i)}) \quad (3)$$

$$\phi \sim \text{Dirichlet}(\beta) \quad (4)$$

$$w_i | z_i, \phi \sim \text{Multinomial}(\phi_{z_i}), \quad (5)$$

where α and β are hyper-parameters for the document-topic (θ) and topic-word (ϕ) Dirichlet distributions, respectively. For simplicity α and β are assumed to be scalars, resulting in symmetric Dirichlet priors.

Given our observed words \mathbf{w} , the key task in LDA is the inference of the hidden topics \mathbf{z} $p(z|w)$.

Unfortunately, this posterior is intractable [9][27][51] and we should resort to estimations. As said before, [27] uses a Markov Chain Monte Carlo sampling scheme, also called ‘‘Collapsed Gibbs Sampling’’. The rationale behind Gibbs sampling for parameter estimation is that instead of directly estimating the topic-word $p(w|z)$ and document-topic $p(z|d)$ distributions, we estimate the posterior probability distribution over latent variable z given the observed data conditioned on the topic assignment for all other word tokens using the below equation (Gibbs Sampling):

$$p(z_i = v | \mathbf{z}_{-i}, \mathbf{w}, \alpha, \beta) \propto \left(\frac{n_{-i,v}^{(d)} + \alpha}{\sum_u (n_{-i,u}^{(d)} + \alpha)} \right) \left(\frac{n_{-i,v}^{(w_i)} + \beta}{\sum_{w'} (\beta + n_{-i,v}^{(w')})} \right) \quad (6)$$

where $n_{-i,v}^{(d)}$ is the number of times topic v is used in document d , and $n_{-i,v}^{(w_i)}$ is the number of times word w_i is generated by topic v . The $-i$ notation signifies that the counts are taken omitting the value of z_i .

Incorporating knowledge into LDA This section describes a mechanism called ‘‘Topic-in-Set knowledge’’ introduced in [3] and [4] for adding ‘‘partial’’ supervision to LDA, in a sense, to incorporating knowledge into LDA.

Traditionally, topic assignments have been denoted by the variable z in

LDA, and these are called “ z -labels” in a set of works [3] [4]. In simpler words, each topic has a label $(0,1,2,3,4,\dots)$. In particular, a z -label gives the knowledge that the topic assignment for a given word position is within a subset of topics [3]. This method is also called “ z -label” LDA. Topic-in-Set Knowledge [3] allows the user to specify a z -label for each observed word in the corpus. This gives the ability to force certain words and the words that are co-occurring the most with them to appear only in certain topics.

A z -label for observed word w_i consists of a set $C^{(i)}$ of possible values for the corresponding latent topic index z_i . From (6) let:

$$q_{iv} = \left(\frac{n_{-i,v}^{(d)} + \alpha}{\sum_u n_{-i,u}^{(d)} + \alpha} \right) \left(\frac{n_{-i,v}^{(w_i)} + \beta}{\sum_{w'} (\beta + n_{-i,v}^{(w')})} \right). \quad (7)$$

We can now easily set a hard constraint by modifying the Gibbs sampling equation with an indicator function $\delta(v \in C^{(i)})$, which takes on value 1 if $v \in C^{(i)}$ and is 0 otherwise. Then we have:

$$p(z_i = v | \mathbf{z}_{-i}, \mathbf{w}, \alpha, \beta) \propto q_{iv} \delta(v \in C^{(i)}) \quad (8)$$

In simple words, this means that a sampled word is added to a topic only if it has co-occurred with one of the seed words in that topic. This formulation gives a flexible method for inserting prior domain knowledge into the inference of latent topics. In simple words, the user can provide seed words for a concept of interest, which are then used to learn a topic built around the concept.

Discussion: In general, “ z -label” LDA and the normal LDA work exactly the same to discover topics. However, they differ in the way they sample words for each topic. Normal LDA allows any words to be sampled for any topics. In contrast, “ z -label” LDA allows a word to be sampled and added to a specific topic only if the word being sampled has co-occurred with one or more of the provided seed words. This will allow each topic to contain less general words, which in turn, makes them more specific.

3.1.2 Ontology Population

As explained in the previous section, now we have a mechanism that can learn a topic built around a concept. Now by having the concepts from our Ontology (6 concepts), and having for each concept a set of “seed” words¹⁸, we can learn six topics. With each topic containing words co-occurring most with the seed words and accordingly around a certain concept. Each of these topics found (list of words) is then added to our Ontology as instances of the corresponding concepts. This way we have populated our Ontology.

The result of this method is two matrices. The ϕ matrix containing the probability of each word given each concept(topic) and the θ matrix containing the probability of each sentence given each concept (topic).

3.1.3 Concept recognition

The next step in this phase is recognizing the concept of a new sentence by comparing it to each of the concepts in the Ontology. Using the ϕ matrix, we calculate p_w^i , the probability of word w belonging to concept i . Suppose the j 'th word is w_j . With the assumption that words are independent given the topic, the probability of a sentence belonging to a concept is just:

$$\prod_j p_{w_j}^i \tag{9}$$

The concept with the highest probability among other concepts is chosen as the candidate concept. This method is used to recognize the concepts of the sentences in our training set. As mentioned before, our idea is to include these concepts in our learning process. In the next section we describe how this is simply done.

3.2 Sentiment Classification

3.2.1 Construction of Feature Vectors

Similar to all vector-space machine learning methods, we need to form our feature vectors. We made a “bag-of-words” feature vectors for each of the sentences in our corpus. Although this will cause our feature vectors to become extremely sparse, we will show that our Active Learning algorithm can handle it perfectly. Also, we use Support Vector Machines (SVM) [10] as our base classifier which is proven to handle high dimensional and sparse

¹⁸Our choice for seed words is detailed in the next chapter

data very well [10][33].

To include the concepts into the learning process, we took an unconventional but simple and effective step. We added six extra dimensions to the bag-of-words feature vectors, each dimension corresponding to a concept (leaf nodes) in our ontology. That is:

$$sentence \rightarrow \underbrace{[\text{bag-of-words} + \text{concept dimensions}]}_{\text{feature vector}} \quad (10)$$

For example, if the 3rd dimension of these additional dimensions corresponds to the concept “*Hoofdgerecht*” and if the recognized concept of a sentence is also “*Hoofdgerecht*”, then this 3rd dimension takes the value of 1 and all other five dimensions get a value of 0. For a detailed discussion about why we believe this works see section 5.2.3. After the construction of feature vectors, they are used in our Active Learning algorithm which is described in the next section. Figure 2 is gives a more clear picture of this process.

3.2.2 Active Learning

The key idea behind *active learning* is that a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns. An active learner may pose *queries*, usually in the form of unlabeled data instances to be labeled by an oracle (e.g., a human annotator). Active learning is well-motivated in many modern machine learning problems, where unlabeled data may be abundant or easily obtained, but labels are difficult, time-consuming, or expensive to obtain [50][58][18].

There are several scenarios in which active learners may pose queries, and there are also several different query strategies that have been used to decide which instances are most informative. The reader is encouraged to refer to [50] for a complete survey of active learning and its query strategies.

Perhaps the most common scenario for many real-world learning problems in which large collections of unlabeled data can be gathered at once is “*pool-based sampling*” or “*selective sampling*” [37], which assumes that there is a small set of labeled data L and a large pool of unlabeled data U available. Queries are selectively drawn from the pool. Typically, instances are queried

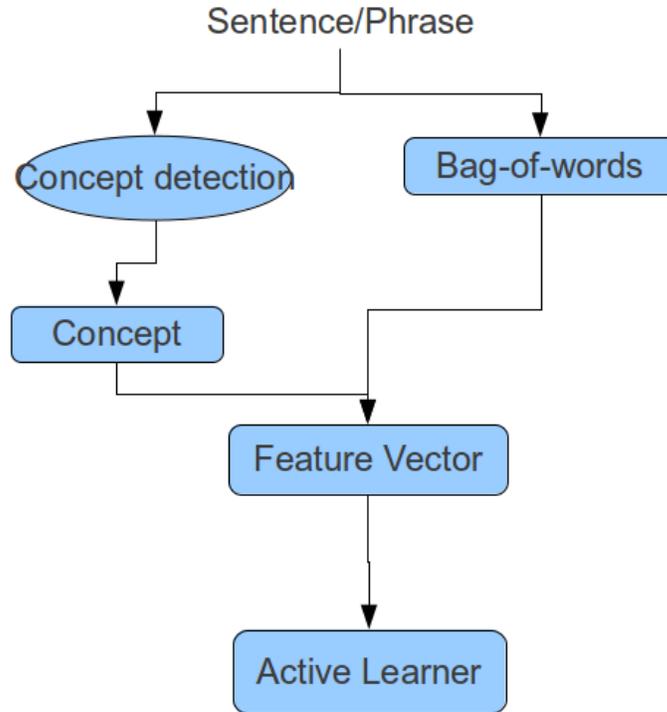


Figure 2: Feature vector construction diagram

in a greedy fashion, according to an informativeness measure used to evaluate all instances in the pool (or, perhaps if U is very large, some subsample of it) [50].

All active learning scenarios involve evaluating the informativeness of unlabeled instances, which can either be generated afresh or sampled from a given distribution. There have been many proposed ways of formulating such query strategies in the literature [50].

Among all query strategies perhaps the simplest and most commonly used query framework is “uncertainty sampling” [37]. In this framework, an active learner queries the instances about which it is least certain how to label. This approach is often straightforward for probabilistic learning models which actually suits our problem. For example, when using a probabilistic model for binary classification, in our case classifying a sentence as either

negative or *positive*, uncertainty sampling simply queries the instance whose posterior probability of being positive is nearest to, for example 0.5.

As mentioned before, our aim is to utilize SVMs as our base classifier in our active learning algorithm. Therefore, uncertainty sampling strategy for SVMs involves querying the instances closest to the linear decision boundary [52].

Having what we said above, our Active Learning algorithm is as follow:

Input L : a small set of labeled datapoints
 U : a set of unlabeled datapoints
 F : a classifier(SVM)
 q : Number of queries that should be asked from the user
 T : Uncertainty threshold

```
while (Continuation_ Criteria) do  
  Train  $F$  on  $L$   
  Using  $F$ , classify  $U$  and get uncertainty scores for each  
   $QU \leftarrow$  select datapoints closest to the threshold  $T$ ,  $n \leftarrow 1$   
  for datapoint in  $QU$  do  
    if  $n \leq q$  then  
       $label \leftarrow$  ask label from user  
       $L \leftarrow (datapoint, label)$   
       $n++$   
    end if  
  end for  
end while
```

The results for our algorithm is presented in the next chapter.

3.3 Opinion Summarization

As mentioned in the introduction, our final output is the summarization of opinions towards the concepts in our Ontology. After the concept detection and sentiment classification phase, each of the sentences is labeled with

its recognized concept along with its sentiment polarity. Then the summarization is achieved by exploiting the hierarchy between the concepts in our domain ontology which enables us to group the sentences and phrases based on their concepts and to produce a summary based on the number of positive and negative opinions for each concept. Moreover, the levels or hierarchy of concepts allows us to produce summaries in different granularities. For example, in our domain ontology “*Food*” is a higher level concept than “*Drank, Voorgerecht, Hoofdgerecht*” and “*Dessert*”. Therefore, based on needs, the summarization of opinions can be done on the “*Food*” level in general along with “*Bediening*” and “*Sfeerkenmerken*”. The workflow of our system to produce these summaries as final outputs is shown in Figure 3.

An example of an output is given in the results chapter.

Discussion: We have to mention that the use of a domain ontology was not necessary for sentiment classifications. We could have trained the modified LDA to discover 6 topics, each around our seed words, and then continue with the feature vector construction and sentiment classification. However, since we needed to produce opinion summaries as the final output of our system, we mainly used an ontology to exploit its hierarchy to produce opinion summaries. Our populated ontology together with the ϕ matrix (which holds probabilities for each word in the ontology given its concept in the ontology) will allow ease of portability to other Dutch websites that are about restaurant reviews in possible future usage.

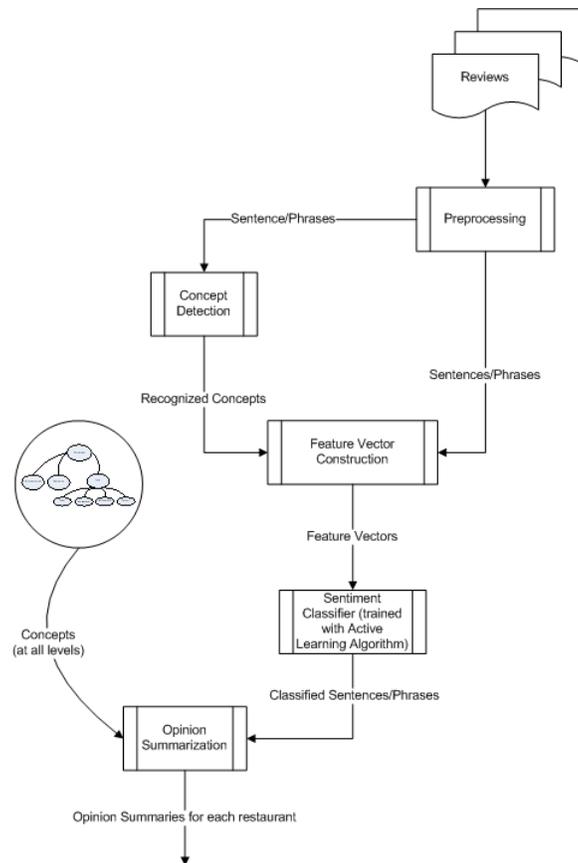


Figure 3: The workflow of our system for producing opinion summaries as final outputs

4 Experimental Setup

In this section we give details about our experimental setup.

4.1 The restaurant review domain dataset

As mentioned a few times in the previous chapters, our corpus consists of restaurant reviews about all restaurants in the city of Amsterdam obtained from <http://www.iens.nl>.

Our initial assumption was that a sentence is about only one concept, and

No. of restaurants	No. of reviews	No. of sentences and phrases
2147	40071	300417

Table 1: Corpus Statistics

reviewers tend to give opinions about different concepts in separate sentences. However, by further observation of the reviews, we saw that usually if the user is giving opinions about 2 or more concepts, he mostly uses a comma “,” to divide the sentence. For example, “*Leuke bediening, eerlijke tent.*” (nice service, honest place). Therefore, we added a simple rule to our sentence tokenizer (step 2 of preprocessing which is discussed next) to tokenize sentences not only based on “.” but also on “,” wherever applicable, which resulted in dividing reviews into sentences and phrases. The statistics of the corpus can be found in Table 1.

4.2 Preprocessing

The preprocessing phase of our corpus consists of the following steps:

1. As we noticed there were substantial amounts of reviews in English. Therefore we developed two N-gram language models[49], one for English (trained on Reuters-21578 Corpus¹⁹) and one for Dutch (trained on CONLL2002 Dutch corpus²⁰) to detect the languages of reviews and discard the English ones.
2. The next step was the review tokenization to divide each review into phrases and sentences.
3. Stop words²¹ were removed.
4. The words were stemmed using snowball’s Dutch stemming algorithm²².

4.3 Training, Development and Test sets

To better show the effectiveness of our approach, we decided to do evaluations for each phase separately. This required careful attention in choosing the test set, development set and training set for each phase.

¹⁹<http://www.daviddlewis.com/resources/testcollections/reuters21578/>

²⁰<http://www.cnts.ua.ac.be/conll2002/>

²¹Snowball’s stop word list: <http://snowball.tartarus.org/algorithms/dutch/stop.txt>

²²<http://snowball.tartarus.org/algorithms/dutch/stemmer.html>

Concept	Count
Drank	174
Voorgerecht	52
Hoofdgerecht	309
Dessert	36
Bediening	242
Sfeerkenmerken	172
Sentiments	Count
Positive	679
Negative	306

Table 2: Corpus sample statistics: number of times that each concept is used in the sample and the number of positive and negative sentiments

The most important part was the evaluation of our sentiment classification phase which shows the effectiveness of importing the concepts into the learning process. As we wanted to compare this phase with a fully supervised method to show this effectiveness, and as this required manual labeling of the sentences and phrases, we had no choice but to choose a small sample of the obtained corpus.

We were able to label 1756 sentences and phrases. Human experts were asked to label each sentence or phrase as having positive, negative or neutral sentiments. In addition, the experts were asked to identify the concept to which the sentence is referring to (corresponding to the six concepts discussed previously) and labeling them with the identified concepts. All other sentences or phrases were labeled as being unrelated (not related to any of the concepts) or having mixed sentiments or talking about more than one concepts. After discarding neutral, unrelated and mixed phrases (mixed in terms of concepts or sentiments) or sentences, about 985 datapoints were left (See Table 2 for the distribution of concepts and sentiments in this subset), of which 20% was allotted for development set, 20% for the test set, and the rest for training set.

We used the same test set to evaluate both the concept detection phase (considering only the concept labels) and sentiment classification phase (considering only the sentiment labels).

Drank	Voorgerecht	Hoofdgerecht	Dessert	Sfeerkenmerken	Bediening
Wijn, bier, koffie, thee	antipasti, soep, amuse, salade	eten,vis, vlees, pizza	nagerecht, taart,ijs, vla	sfeer, fancy, ro- mantisch, decor	personeel, service, gedrag, houding

Table 3: A short list of seed words used for each concept

4.4 Concept detection

After the preprocessing operations, at this point, the sentences are represented as “bag-of-words” and these “bag-of-words” are fed to the concept detection module. As this phase is completely independent of the next phase, we trained our modified LDA on the whole corpus, excluding the 1756 datapoints mentioned above to avoid overlaps. We set the algorithm to learn 6 topics corresponding to the concepts in our ontology. For each concept, we provided several seeds. An example of the seed lists²³ for each concept can be found in Table 3.

Our choice of seed words was very simple. We simply chose a few synonyms for each concept, along with some examples of that concept. We also added a few related words to each concept. For this, the concepts were given to a native Dutch speaker and asked to write the first few words that came to his mind about each concept. For example, for the concept “Bediening”, words such as *houding* (attitude) and *gedrag* (behavior) were proposed.

We used protege [35] to build our Ontology. The results on the test set are presented in the next chapter.

4.5 Sentiment Classification

At this point, after construction of the feature vectors as we discussed in section 3.2.1, they are given to our Active Learning algorithm as Inputs. In our Active Learning algorithm for Sentiment Classification, we used a Linear Kernel for the SVM classifier and we set the maximum number of data points to be asked from the user for labeling in each iteration of the algorithm to 10 (in practice this was less in some iterations, due to the number of uncertain points²⁴ that fall into the uncertainty threshold). In addition,

²³A complete list of all seed words can be found the in the appendix of this paper

²⁴The libSVM classifier package which we use is able to assign an uncertainty score to each of the datapoints being classified

the number of learning iterations was set to a maximum of 10. However, in practice the algorithm was converging before reaching the maximum number of iterations.

Following our Active Learning algorithm, we needed a small set of labeled datapoints as seeds to start with. Therefore, we chose 30 datapoints from the labeled training set mentioned at the end of section 4.3 and we discarded the labels for the rest of the datapoints, considering them as unlabeled datapoints set U in our algorithm.

The results of our active learner are presented in the next chapter.

4.6 Supervised method(base-line)

Following what we mentioned about the evaluations in section 4.3, we trained an SVM classifier with a linear kernel on the training data using only the bag-of-words for the feature vectors and only the sentiment labels. This way we can easily observe the effect of the inclusion of concepts as extra dimensions in the feature vectors. For this we did our evaluations on the same test set as we used for our own approach. The results are presented in the next chapter along with a comparison with our active learning approach.

5 Results and Analysis:

This chapter presents the results of our approach. First the results of the Concept detection phase, its accuracy and ability to recognize the concepts of sentences and phrases to which they refer (with some examples) is presented.

Next, the results for the Sentiment Classification phase are given. The results of this phase are evaluated from different aspects. First, a comparison of the results of our Active Learning algorithm with different decision boundaries is presented. Next, our method is compared to our baseline (mentioned in 4.6) in terms of accuracy, inclusion of concepts in the feature vectors and the number of labeled datapoints needed for each method to achieve such accuracies. After that, a few classification examples are given. Finally, an example of the generated opinion summary is presented.

The results of both phases are also analyzed separately in detail in their corresponding section.

5.1 Concept detection results and analysis

As mentioned before, we trained our modified LDA on the training corpus to discover six topics, each around a certain concept using some seed words. Each topic contains a list of most probable words generated from that topic. Table 4 shows the most probable words for each topic (concept).

Notice that these are the stemmed words as we did stemming for our pre-processing. Ignoring the seed words for each concept (mentioned in the appendix) from these lists, obviously we can see a lot of related words showing up for each concept. This is because of the modified Gibbs sampling scheme that forces certain words and their co-occurrences to only show up in certain topics. This works perfectly fine if there is a set of words that substantially co-occur with one or more of the seed words.

In other words, if there is a concept that is widely discussed throughout the review texts, LDA is better able to discover its corresponding topic from the text. In contrast, if the concept is not widely discussed, then LDA is unable to discover a topic for that concept. Our preliminary word-occurrence statistics²⁵ show that the most frequently used words belong to the concepts of “*Drank*”, “*Hoofdgerecht*”, “*Bediening*” and “*Sfeerkenmerken*” which means

²⁵For this we made a very long list of most frequent words(excluding the stopwords)

Topic(Alphabetical order)	Top words
Bediening	bedien vriendelijk goed erg personel zer aardig hel servic wel leuk attent slecht echt snel avond prima jammer vlot mens eigenar druk uitstek prettig restaurant gastvrij top kok best bijzonder gast super herhal zak vatbar maakt ervar behulpzaam aandacht verder vond jong
Dessert	geget kom ker zeker heerlijk terug wer aanrader restaurant gan echt jar ga wek lekker avond afgelopen nooit dessert gister eerst laatst vriend gehad waard snel absolut vorig geled vaker kas twee kortom par zaterdag grag nagerecht regelmat vak toetj volgend tijd ieder prober elk taart
Drank	wijn kaart goed menu euro wel lekker koffie klein gang heerlijk grot porties drink keuz had erg glas huiswijn person per allen dur fles uitgebreid besteld reken prima hel kreg jammer ruim mooi wit wijnkaart twee water rod bestell echt verrass huis verschill menukaart zer genoeg glaz glaasj ros
Hoofdgerecht	eten goed lekker gerecht prijs kwaliteit heerlijk hoofdgerecht echt prima keuk erg vles zer smak wel verhoud redelijk hel ver vis uitstek pizza prijs pasta lunch bijzonder gewon restaurant hog kip aanrader dur kaart bereid slecht verwacht italiaan vond weinig maaltijd matig zeker war smakelijk mooi ingredient vind eerlijk beter smaakt ok geget geld goedkop betal
Sfeerkenmerken	restaurant gezell sfer tafel leuk mooi amsterdam goed zit erg echt wel hel war tent inricht bar terras lekker zak interieur plek avond locatie best prachtig beter vind buurt klein prima muziek ziet buit grot binn ambianc jammer eetcaf ingericht uitzicht fijn beetj aanrader druk top caf gewon ruimt geweld wer lokatie
Voorgerecht	lekker heerlijk voorgerecht smak salad erg ver goed echt vooral vooraf brod groent saus wel aanrader gebak soep smaakt besteld geserveerd had drog zoal kreg tonijn zalm koud bord vet nam daarna beetj rijst garnal sat allen hel warm sla friet gevuld zat zout mal rod biefstuk gegrild gamba broodj

Table 4: Top words in each topic

that reviewers mostly talk about these concepts in their review. Moreover, the statistics from the chosen sample of the corpus in Table 2 bear witness to this fact. The reflection of this phenomena can be seen in the Table 4. Topics such as, “*Voorgerecht*” and “*Dessert*” are among the ones which are

Concepts	Accuracy(%)
Drank	85.0
Voorgerecht	52.5
Hoofdgerecht	85.0
Dessert	40.0
Bediening	82.1
Sfeerkenmerken	71.0

Table 5: Results of modified LDA on the test set

less discussed in the reviews. Therefore, our modified LDA is unable to discover enough proper words around these concepts, resulting in a list of words with many (semantically) unrelated members. In all other cases, our modified LDA was capable of discovering a decent amount of related words for each topic.

Using these discovered topics and following the approach which we discussed in section 3.1.3 we detected the concepts of the sentences and phrases in our test set. The results can be seen in Table 5. The evaluations are for each concept separately.

What we discussed before about the prevalence of discussion around a concept in the reviews is also evident here. The least accuracies belong to the concepts “*Dessert*” and “*Voorgerecht*”, and this is due to the inability of the LDA to discover proper words for these topics which resulted in an amount of unrelated words for these concepts, which makes the recognition of the correct concept extremely difficult and error prone.

To see if extending the list of seeds for these concepts would improve the results, we added several extra seed words for these two concepts. However, the results showed no improvement. Our investigations on the review corpus showed that these extra seed words were almost never used or very rarely used, which made the extension of the seed-word list ineffective. A larger corpus would probably have a more diverse use of words and could be the solution.

In Section 1.1 we briefly talked about “explicit features” and “implicit

no	Phrase	Recognized as:
1	Alle jongens en meiden die ons hebben geholpen (en dat waren er best veel) waren gastvrij	Bediening
2	plichtmatig en alles behalve spontaan	Bediening
3	tafeltje gereserveerd en we werden zeer vriendelijk ontvangen	Bediening
4	De inrichting maakt wel het eea goed	Sfeerkenmerken
5	vooral de lamsschenkel was boven verwachting goed	Hoofdgerecht
6	Leuk van de nederlandse wijn	Drank
7	entrecote besteld: kleine portionering voor 20 euro	Hoofdgerecht
8	De hapjes echter te klein	Voorgerecht

Table 6: A few examples for which the concepts where correctly recognized

no	Phrase	Recognized as:
1	Maar als het vervolgens een kwartier duurt voor het personeel je opmerkt	Dessert
2	De mezzes zijn een ware traktatie	Drank
3	Afgelopen zaterdag voor het eerst bij de Kas gegeten en het was heerlijk: lekker eten	Dessert
4	Waar ik minder over te spreken ben is de ambiance	Bediening

Table 7: A few examples for which the concepts where incorrectly recognized

features” and we claimed that our method is able to recognize *implicit features* also. Table 6 (for English translations refer to Appendix A.2) shows actual examples and their recognized concepts from the test set. For example, obviously in the sentence “*tafeltje gereserveerd en we werden zeer vriendelijk ontvangen*” (reserved a table and we were very kindly received) there is no indication of “*Bediening*” or “*Service*”. Moreover, none of the seed words in our seed-word list can be seen in these sentences which again bears witness to the strength of our method.

Table 7 show some example that concept detection phase was not able to correctly recognize their concepts (For English translations see Appendix A.2).

Unfortunately, because of the aforementioned problems (concepts “*Dessert*” and “*Voorgerecht*” being less discussed in the reviews), we observed that topics “*Dessert*” and “*Voorgerecht*” have turned into topics for harbouring

words that either could not go into other topics or they have turned up with very low probabilities in other topics. This, in combination with our rather naive assumption which we described in Section 3.1.3 ²⁶ is the main reason for the concepts of these sentences and phrases to be incorrectly recognized. For example, eventhough phrase No.1 has a very high probable word *personeel* for the topic “*Bediening*”, its concept is still incorrectly recognized. The reason is that words such as “*vervolgens, kwartier*” and “*opmerkt*” have been wrongfully assigned to the topic “*Dessert*” which shift the topic distribution of the sentence towards the topic “*Dessert*”.

Having said above, we can claim that our concept detection phase works decent as long as the concepts which are going to be discovered are relatively widely discussed in the reviews, which is possible by selecting a larger training corpus. This is completely reasonable since our method is unsupervised (except providing seeds for each concept) and obtaining large amounts of reviews from the Internet is cheap since also no labeling is needed.

5.2 Sentiment Classification phase results and analysis

Following our approach, we included the concepts (from the concept detection phase) in our feature vectors on which we trained our Active Learning algorithm. We set the maximum number of training iterations to 10 and the number of datapoints to be asked from the user for labeling to 10 also.

5.2.1 Results for different decision boundaries

Remember that we used SVM as our base classifier in our Active Learning algorithm (using uncertainty sampling strategy). Common SVM classifier packages²⁷ are able to assign probabilities²⁸ to classified datapoints. In the case of a binary classification (e.g., classification between A and B), the decision boundary in terms of probabilities is 0.5.

Uncertainty sampling strategy for SVMs involves querying the instances closest to the linear decision boundary. To find the best region around this

²⁶We assumed that words in a sentence are independently drawn from different topic distributions

²⁷For example, libSVM <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

²⁸In some papers they are called uncertainty scores

No	Decision Region	Average accuracy(%) $\pm \sigma$
1	0.475 – 0.525	80.11 \pm 2.1
2	0.45 – 0.55	82.96 \pm 2.70
3	0.40 – 0.60	74.0 \pm 5.3

Table 8: Results for Sentiment Classification

boundary, we did several experiments. We mainly sampled datapoints in the uncertainty regions: 0.45 – 0.55, 0.475 – 0.525 and 0.40 – 0.60. For example, imagine that we set the uncertainty region to be 0.45 – 0.55. This means that if the SVM classifier assigns an uncertainty score of 0.52 to a certain datapoint, this datapoint falls into the uncertainty region and can be sampled. It should be added that since our uncertainty sampling strategy randomly chooses the datapoints within the designated boundary, the results for each complete run of the algorithm was slightly different. Therefore, we ran our algorithm 10 times for each of the experiments and we took mean accuracy of these 10 runs along with the calculation of the standard deviation σ . The results can be seen in Table 8.

As can be seen from Table 8 the best results belong to our second experiment. Although the first experiment seems to be a bit more stable (having a slightly lower σ), its classification accuracy is lower than the 2nd experiment. Moreover, the difference between σ 's of the experiments 1 and 2 is very small and negligible. The slightly better stability in experiment 1 in comparison to experiments 2 and 3 is a result of a tighter sampling boundary. By looking at the datapoints queried by the oracle (within this boundary), we observed that almost the same datapoints tend to be sampled in each of the 10 runs. Although tightening the boundary brings more stability to our algorithm, it will cause the accuracy to drop, and this is because that tighter sampling boundary limits the diversity (in terms of diversity of words) of datapoints being added to the training set in each iteration and this will result in the addition of less knowledge to the classifier. However, this does not mean that a wider sampling boundary necessarily improves the classifier's accuracy.

The results for experiment 3 bear witness to the fact that a very large sampling boundary can indeed be harmful to classifier's accuracy, not to mention increasing its instability.

Method	Average(%)	no. of labeled data (Incl. seeds)
Our approach	82.96	102 (average for 10 runs)
Baseline	74.9	591

Table 9: Comparison of results with base-line

5.2.2 Results in terms of number of labeled datapoints

In terms of the number of labeled datapoints, the difference is very obvious (see 3rd column of Table 9). Our method could achieve a much higher accuracy using a lot less labeled datapoints. This phenomena can also be justified as follows: One of the strengths of our Active Learning algorithm is that the SVM classifier is somehow being guided by the labeling of the user in each iteration (within the boundary). This is in fact the winner card for Active Learner in comparison to a fully supervised learner. By allowing the Active Learner to sample from anywhere without limiting it to a sampling boundary until no unlabeled datapoint is left, the results of the Active Learner almost equals to those of a fully supervised classifier.

5.2.3 Results in terms of inclusion of concepts into feature vectors

As it can be seen in Table 9, our Sentiment Classification method outperforms the baseline. We tested for significant differences, mentioned in [22], between the results of our own method and the base-line and the results show that this difference is indeed significant. To see if this difference between the accuracy of both methods is because of the inclusion of concepts directly into the feature vectors (by adding extra dimensions), we did an extensive study on the test set and did a comparison between the results of the baseline and our method. Several examples of the test set can be seen in Table 10 (See Appendix A.2 for English translations). Each row consists of the phrase or sentence being classified, its concept which is correctly recognized by the concept detection phase, the sentiment label assigned by our approach, the sentiment label assigned by the baseline approach and finally the true label.

No.	Phrase	Concept	Our Method	Baseline	True label
1	entrecote besteld: kleine portionering voor 20 euro	Hoofdgerecht	Neg.	Neg.	Neg.
2	tafelje geserveerd en we werden zeer vriendelijk ontvangen	Bediening	Pos.	Pos.	Pos.
3	kleine en gezellige tent	Sfeerkenmerken	Pos.	Neg.	Pos.
4	geen stoffig traditioneel interieur maar fris en eigentijds	Sfeerkenmerken	Pos.	Neg.	Pos.
5	wat anders werd je ongevraagd betrokken bij de ruzies tussen de bediening en de keuken	Bediening	Pos.	Pos.	Neg.
6	Dit is nu voedsel voor je lichaam en geest	Hoofdgerecht	Neg.	Neg.	Pos.

Table 10: Classification comparison examples(Pos:Positive,Neg:Negative)

Examples 1 and 2 are among the phrases that were correctly classified by both methods. *“kleine”* and *“vriendelijk”* seem to be good indicators of negative sentiment (for *“Hoofdgerecht”* concept) and positive sentiment(for *“Bediening”* concept) respectively. In contrast, the word *“kleine”* in example 3 indicates a positive sentiment towards the concept *“Sfeerkenmerken”*. The difference is subtle here and comparison may seem a bit controversial.

By the delving into the training and test set, we found out that, overall, number of times that the word *“kleine”* was used as a negative sentiment is much more than (252 times) the number of times the *“gezellige”* was used to indicate positive sentiment (8 times) in the training set. For this reason the SVM classifier in the base-line treat the words *“kleine”* as a negative indicator all over the test set.

While our approach distinguishes between the word *“kleine”* when it is used in the context of *“Hoofdgerecht”* and when it is used in the *“Sfeerkenmerken”* context. We observed that, one the one hand, the number of times that the word *“kleine”* was used in the context of *“Sfeerkenmerken”* to indicate a positive sentiment (61 times) was much more than when it was used to indicate a negative sentiment in the same context (13 times). Therefore, our approach

classifies “*kleine*” as a positive indicator in the context of “*Sfeerkenmerken*”.

On the other hand, the number of times that this word was used in the context of “*Hoofdgerecht*” to indicate a negative sentiment (169 times) was much more than its use to indicate a positive sentiment in this context (29 times). Therefore, our approach classifies “*kleine*” as a negative indicator in the context of “*Hoofdgerecht*”. More importantly, our observations show that the word “*gezellige*” was used only a few times in the context of “*Sfeerkenmerken*” (3 times), therefore its effect is much less than the word “*kleine*” to change the sentiment polarity of the phrase.

This is why our classifier is able to assign the correct sentiment label to phrase no.3 and this is because of the effect of including the concepts in the feature vectors which results in the changing of sentiment polarities according to a change in the context. On the other hand, the word “*kleine*” was almost all the time being used to indicate a negative sentiment towards the “*Hoofdgerecht*”²⁹. However, in this case (example 1) both methods were able to do the correct classification.

Example 2 is also a very good example of our approach’s ability to do sentiment classification on “implicit” features, and example 4 for “explicit” features.

Examples 6 and 7 are representatives of very complex and difficult phrases to classify for both methods. Although our concept detection module was able to correctly recognize their concepts, the inclusion of these concepts into the feature vectors was ineffective. We noticed that these sentences are mostly composed of words with extremely low frequencies which makes them specially difficult to classify. Most probably a larger training corpus might help.

Discussion: Why it works? In general, a combination of context, sentence words and sentiment is needed for a correct sentiment classification. If we do not include concepts then a word in a sentence will be treated, by the SVM classifier, in the same way for all classes (positive or negative).

²⁹In addition, there were several cases that it was used to indicate a negative sentiment towards “*Voorgerecht*” and “*Dessert*”

However, in this way we classify a pair <phrase, concept> as a positive or negative instead of only classifying a phrase as having positive or negative sentiment. The results and their analysis show that indeed our approach works well and can outperform the baseline method and were able to increase the sentence-level sentiment classification accuracy by including the concepts in the feature vectors.

Discussion As we saw from the results and the analysis, on the one hand, our concept detection method fails to recognize certain concepts when these concepts are not widely discussed. In other words, when the concepts are not evenly discussed (distributed). This will result in the propagation of errors into the sentiment classification phase (by inserting wrongly recognized concepts into the feature vectors). On the other hand, eventhough we randomly chose a small sample of the whole corpus and eventhough the results of our approach are significantly better than the results from the baseline approach, this sampled set is still a bit small to be a complete representative of the whole corpus and makes our approach sensitive to the distribution of the dataset on which is being evaluated (test-set). As mentioned before, because we needed to compare our approach with a supervised method, we had no choice but to select a small sample of corpus for labeling. However, we should mention that this problem (or weakness) is not only specific to our approach, but it is a common problem in machine learning approaches when there is a lack of datapoints, not to mention the computational resources.

5.3 Opinion Summarization

So far, we were able to recognize the concepts of sentences and phrases and we could determine their sentiment polarity based on the context of each sentence or phrase. After this, sentences and phrases in the reviews for each restaurant are tagged with their corresponding concepts and sentiment polarities.

Our final output is the production of an opinion summary of these sentences and phrases for each restaurant. In other words, we are able to produce a summary of opinions for each restaurant, reporting what do people think about the concepts “*Sfeerkenmerken*”, “*Hoodfgerecht*” and etc. in terms of positive or negative sentiments and how many positive or negative sentiments have been mentioned towards each of these concepts.

This is achieved by exploiting the hierarchy between the concepts in our

domain ontology which enables us to group the sentences and phrases based on their concepts, to count the number of positive and negative opinions for each concept, and produce an opinion summary in the end. An example is seen in Figure.4.

```

Restaurant name: Eetcafé 't Pakhuis
{
  Hoofdgerecht:{
    positive: {
      count: 134
      id of phrases in the corpus: 234,456,477,...
    }
    negative: {
      count: 65
      id of phrases in the corpus: 250,262,273,...
    }
  }
  Bediening:{
    positive: {
      count: 123
      id of phrases in the corpus: 240,257,...
    }
    negative: {
      count: 12
      id of phrases in the corpus: 251,260,276,...
    }
  }
  .
  .
  .
}

```

Figure 4: A partial example of an opinion summary

Moreover, the levels or hierarchy of concepts allows us to produce summaries in different granularities. For example, in our domain ontology “*Food*” is a higher level concept than “*Drank, Voorgerecht, Hoofdgerecht*” and “*Dessert*”. Therefore, based on needs, the summarization of opinions can be done on the “*Food*” level in general along with “*Bediening*” and “*Sfeerkenmerken*”.

6 Future work

In addition to using a larger training corpus for both phases, the use of certain Natural Language Processing methods can be beneficial in the future developments. For example, for a better and more accurate division of sentences into phrases when there is a concept change within a sentence. These methods can be in the form of very simple to complex syntactical rules.

Moreover, a use of Part-of-Speech Tagger could be beneficial in reducing noisy data. After we decided to divide certain sentences into phrase, as we mentioned before, a lot of phrases were produced that did not have a meaning per se. Most of these so called “phrases” did not contain any Nouns or Adjectives. Almost all of these phrases can be discarded, as phrases without any Noun or Adjective are less likely to carry any sentiments or talk about a certain concept.

7 Conclusion

In conclusion, we think that we have achieved our final goal and we were able to answer our research questions as we were expecting.

In summary the main achievements and main advantages of our project can be enumerated as follows:

1. We were able to recognize the concept to which a sentence or phrase is referring to
2. We were able to improve the sentence-level sentiment classification accuracies by including the concepts directly in to the feature vectors
3. As far as we know, the inclusion of concepts has not been done in the previous works which makes our approach special in performing context based sentiment analysis in a very simple but effective manner
4. Moreover, as far as our studies show, this is also the first attempt on doing sentiment analysis on “implicit” features. Existing works try to identify “explicit” features with NLP methods or other external tools. Our method does not use any of these. Most of the previous works are language dependent, especially when it comes to identify features as they use language dependent tools. However, our approach is completely independent of language.

5. Our approach needs very little supervision (in terms of providing seed words) and in terms of the number of labels needed to achieve a decent accuracy, it works with almost one-fifth of the labeled datapoints used for a supervised method. In addition, with this number of labeled datapoints we could significantly outperform the baseline.

A Appendix

A.1 Seed Lists

Topic(Alphabetical order)	Top words
Bediening	gedrag houding personeel service wacht verget
Dessert	nagerecht dessert taart kaas ijs appelmoes chocolademousse pudding roomijs sundae tiramisu trifle vla
Drank	drinken vruchtensap bier cocktail jenever rose melk mineraalwater sappen thee whiskey wijn champagne koffie thee cognac
Hoofdgerecht	keuken lunch dinner porties maaltijd gang gerecht eten vis vlees kip rijs rijst lasagna pannenkoek pasta pekingeend pizza ravioli spaghetti stamppot
Sfeerkenmerken	sfeer locatie fancy geen muziek hip 'honden toegestaan kindvriendelijk klassieker solo romantisch 'rustig praten' single spot zakelijk faciliteiten airconditioner rolstoeltoegankelijk rookruimte serre speelruimte wifi decor behang tafel interieur behangetje
Voorgerecht	starter amuse antipasti bruschetta carpaccio garnalencocktail 'gevuld ei' 'insalata caprese' 'russisch ei' salade soep

A.2 Table translations to English

no	Phrase	Recognized as:
1	All boys and girls who helped us (and there were quite a few) were hospitable	Service
2	perfunctory and anything but spontaneous	Service
3	reserved a table and we were very kindly received	Service
4	The interior compensates something	Atmosphere
5	especially the lamb shank was better than expected	Main dish
6	Nice Dutch wine.	Drink
7	ordered steak: small portioning for 20 euros	Main dish
8	The food was too small.	Appetizer

Table 11: English translation of Table 6

no	Phrase	Recognized as:
1	But you have to wait another 15 minutes before the personnel notices that you are there	Dessert
2	The mezzes are a real treat	Drink
3	last Saturday at the Greenhouse we ate for the first time and it was delicious: good food	Dessert
4	I'm talking less about the atmosphere	Service

Table 12: English translation of Table 7

No.	Phrase	Concept	Our Method	Baseline	True label
1	ordered steak: small portioning for 20 euros	Main dish	Neg.	Neg.	Neg.
2	reserved a table and we were very kindly received	Service	Pos.	Pos.	Pos.
3	Small and cozy place	Atmosphere	Pos.	Neg.	Pos.
4	No dusty traditional but fresh and contemporary interiors	Atmosphere	Pos.	Neg.	Pos.
5	otherwise you become involved in the unwanted arguments between the service and the kitchen.	Service	Pos.	Pos.	Neg.
6	This is food for your body and mind.	Main dish	Neg.	Neg.	Pos.

Table 13: English translations of Table 10

References

- [1] S. Afantenos, P. Denis, P. Muller, and L. Danlos. Learning recursive segments for discourse parsing. *Arxiv preprint arXiv:1003.5372*, 2010.
- [2] C. Andrieu, N. De Freitas, A. Doucet, and M.I. Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1):5–43, 2003.
- [3] D. Andrzejewski and X. Zhu. Latent dirichlet allocation with topic-in-set knowledge. *Semi-supervised Learning for Natural Language Processing*, page 43, 2009.
- [4] D.M. Andrzejewski. *Incorporating Domain Knowledge in Latent Topic Models*. PhD thesis, UNIVERSITY OF WISCONSIN, 2010.

- [5] N. Asher, F. Benamara, and Y.Y. Mathieu. Appraisal of opinion expressions in discourse. *Linguisticæ Investigationes*, 32(2):279–292, 2009.
- [6] A. Aue and M. Gamon. Customizing sentiment classifiers to new domains: a case study. In *Proceedings of RANLP*, volume 49. Citeseer, 2005.
- [7] M. Becker, W. Drozdzyński, H.U. Krieger, J. Piskorski, U. Schöfer, and F. Xu. Sproutshallow processing with typed feature structures and unification. In *Proceedings of the International Conference on NLP (ICON 2002), Mumbai, India, 2002*.
- [8] S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G.A. Reis, and J. Reynar. Building a sentiment summarizer for local service reviews. In *WWW Workshop on NLP in the Information Explosion Era*, 2008.
- [9] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.
- [10] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.
- [11] A. Cadilhac, F. Benamara, and N. Aussenac-Gilles. Ontolexical resources for feature based opinion mining: a case-study. In *23rd International Conference on Computational Linguistics*, page 77, 2010.
- [12] G. Carenini, R.T. Ng, and E. Zwart. Extracting knowledge from evaluative text. In *Proceedings of the 3rd international conference on Knowledge Capture*, pages 11–18. ACM, 2005.
- [13] X. Cheng and F. Xu. Fine-grained opinion topic and polarity identification. In *Proceedings of LREC*, pages 2710–2714. Citeseer.
- [14] H.L. Chieu and H.T. Ng. Named entity recognition: a maximum entropy approach using global information. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.
- [15] P. Cimiano. *Ontology learning and population from text: algorithms, evaluation and applications*. 2006.
- [16] G. D’Agostini. Bayesian inference in processing experimental data: principles and basic applications. *Reports on Progress in Physics*, 66:1383, 2003.

- [17] S.R. Das and M.Y. Chen. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9):1375–1388, 2007.
- [18] Sanjoy Dasgupta and John Langford. Active learning tutorial. *ICML*, 2009.
- [19] K. Dave, S. Lawrence, and D.M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.
- [20] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.
- [21] A. Devitt and K. Ahmad. Sentiment polarity identification in financial news: A cohesion-based approach. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 984, 2007.
- [22] T.G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998.
- [23] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422. Citeseer, 2006.
- [24] M. Gamon. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on Computational Linguistics*, pages 841–es. Association for Computational Linguistics, 2004.
- [25] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. Pulse: Mining customer opinions from free text. *Advances in Intelligent Data Analysis VI*, pages 121–132, 2005.
- [26] N. Godbole, M. Srinivasaiah, and S. Skiena. Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*. Citeseer, 2007.

- [27] T.L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228, 2004.
- [28] T.R. Gruber et al. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.
- [29] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.
- [30] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.
- [31] V. Jijkoun, M. de Rijke, and W. Weerkamp. Generating focused topic-specific sentiment lexicons. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 585–594. Association for Computational Linguistics, 2010.
- [32] Y. Jo and A.H. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824. ACM, 2011.
- [33] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, pages 137–142, 1998.
- [34] T. Kakkonen, N. Myller, and E. Sutinen. Applying latent dirichlet allocation to automatic essay grading. *Advances in Natural Language Processing*, pages 110–120, 2006.
- [35] H. Knublauch, R.W. Ferguson, N.F. Noy, and M.A. Musen. The protégé owl plugin: An open development environment for semantic web applications. *The Semantic Web-ISWC 2004*, pages 229–243, 2004.
- [36] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 282–289. Citeseer, 2001.
- [37] D.D. Lewis and W.A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR*

- conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.
- [38] F. Li, M. Huang, and X. Zhu. Sentiment analysis with global topics and local dependency. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.
- [39] B. Liu. *Web data mining*. Springer, 2007.
- [40] B. Liu. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, pages 978–1420085921, 2010.
- [41] Q. Mei, X. Ling, M. Wondra, H. Su, and C.X. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. ACM, 2007.
- [42] G.A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
- [43] T. Mullen and N. Collier. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of EMNLP*, volume 4, pages 412–418, 2004.
- [44] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.
- [45] G. Petasis, V. Karkaletsis, G. Paliouras, A. Krithara, and E. Zavitsanos. Ontology population and enrichment: State of the art. *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution*, pages 134–166, 2011.
- [46] A.M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 339–346. Association for Computational Linguistics, 2005.
- [47] G. Qiu, B. Liu, J. Bu, and C. Chen. Expanding domain sentiment lexicon through double propagation. In *International Joint Conference on Artificial Intelligence (IJCAI-09)*, 2009.

- [48] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [49] C. Ramisch. N-gram models for language detection. 2008.
- [50] B. Settles. Active learning literature survey. *Machine Learning*, 2010.
- [51] M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427, 2007.
- [52] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.
- [53] P. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL’02)*, 2002.
- [54] W. Wei, P. Barnaghi, and A. Bargiela. Probabilistic topic models for learning terminological ontologies. *IEEE Transactions on Knowledge and Data Engineering*, pages 1028–1040, 2009.
- [55] C. Whitelaw, N. Garg, and S. Argamon. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 625–631. ACM, 2005.
- [56] T. Zagibalov and J. Carroll. Automatic seed word selection for unsupervised sentiment classification of chinese text. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 1073–1080. Association for Computational Linguistics, 2008.
- [57] L. Zhao and C. Li. Ontology based opinion mining for movie reviews. *Knowledge Science, Engineering and Management*, pages 204–214, 2009.
- [58] X. Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2006.
- [59] L. Zhuang, F. Jing, and X.Y. Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50. ACM, 2006.