Automatic Construction of Domain Specific Sentiment Lexicons for Hungarian

Viktor Hangya

Department of Computer Algorithms and Artificial Intelligence
University of Szeged, Hungary
hangyav@inf.u-szeged.hu

Abstract. Sentiment analysis has become an actively researched area recently, which aims to detect positive and negative opinions in texts. A good indicator for the polarity of a given text is the number of words in it that have positive or negative meanings. The so called sentiment lexicons are lists containing words together with their polarities. In this paper we present methods for creating sentiment lexicons automatically. We use these lexicons in sentiment analysis tasks on general and domain-specific Hungarian corpora. We compare the efficiency of sentiment lexicons from different domains and show the importance of using domain-specific sentiment lexicons for different sentiment analysis tasks.

Keywords: sentiment analysis, sentiment lexicon, natural language processing

1 Introduction

People have the opportunity to share their thoughts with others thanks to the increased popularity of social media. A big amount of data is created daily which contains people's opinions about various topics like products, celebrities and companies. In the past few years sentiment analysis has become popular not only in scientific research but also in economy as well. The task of sentiment analysis is to determine the polarity of an opinion in a given document.

The task can be considered as a classification problem in which documents have to be classified into positive or negative classes according to the opinions they contain. When a big amount of annotated data is available, a good method is to create supervised machine learning based classifiers which rely on words from the documents. But this method can be inaccurate if the size of the annotated corpus is insufficient because word types have low frequency in it. To overcome the lexical sparsity problem, sentiment lexicons can be useful, which contain a predefined set of positive and negative words. This knowledge can be used to extract various features besides n-grams. Many general purpose sentiment lexicons are available for English [1, 2] but there are much fewer for Hungarian.

A simple method for creating lexicons is to translate one from a foreign language. However, this method has some disadvantages. First, the process can be time consuming and expensive. Furthermore, polysemous words can have different polarities in different text domains, thus domain-specific lexicons are needed. Most of the available foreign lexicons are for general use. In this paper, we show that domain-specific lexicons are more useful for sentiment analysis on domain-specific corpora. Furthermore, we propose language independent techniques for creating lexicons automatically. By incorporating texts from a given domain we created lexicons from scratch which are useful for extracting features for sentiment analysis tasks. We also created semi-automatic methods which can extend a given seed lexicon by using word similarity.

We evaluated lexicons by employing them in sentiment analysis tasks. For this we used two Hungarian text databases, one is a domain specific corpus which contains reviews about IT products, the other contains texts from news [3] related to various topics, like sports, politics, etc. We show that it is important to use lexicons from the same domain as the texts which we classify into polarity classes.

2 Sentiment Lexica

The most important indicators of sentiments are sentiment words, which can express positive and negative opinions. Three main approaches exist for creating sentiment lexicons: manual, dictionary-based and corpus-based [4]. The manual approach is time-consuming thus expensive. Dictionaries contains synonym and antonym sets for words. Dictionary-based approaches use this knowledge to automatically collect sentiment words starting from a manually created seed lexicon. The goal of the corpus-based approaches is to employ knowledge which can be found in a set of documents. Seed lexicons can be extended by using rules, e.g. adjectives on both sides of a conjunction in a sentence have the same polarity. If the documents are also labeled with positive and negative labels, statistical methods can be used to create lexicons from scratch. In this work, we propose new methods from all three main approaches.

An important fact in using sentiment lexicons is the domain from which the texts come from because some words can have different polarities in different domains. Consider the following example:

- The usage of this mixer is easy and it is very **silent**.
- For this price it's too **silent** for me, I thought it will be louder.

The first example is from the domain of kitchen devices, where *silent* has a positive meaning. In contrast, the second example is from the speakers domain, where *silent* is a negative quality. From this, it can be seen that in a sentiment analysis task choosing lexicons from the appropriate domain is important. There is a need for an automatic method which can create domain-specific lexicons, because there are no lexicons for every domain and creating them manually is expensive and requires an expert in that domain.

In the following we present methods for creating and adapting sentiment lexicons and also highlight their positive and negative aspects.

2.1 Translating a Foreign Lexicon

In our experiments we used a manually translated lexicon for comparison reasons. In English there are many general purpose lexicons, i.e. SentiWordNet [1] or MPQA [2]. We had access to an English lexicon already used by a reputation monitoring system, which we translated to Hungarian. The translated lexicon contained 3322 word forms each with its polarity level from the $[-5, 5]$ interval, where -5 is the most negative value and 5 is the most positive. The translation was carried out manually and we used all of the possible Hungarian translations of a given word.

The method has some disadvantages. First, translation can be time-consuming thus expensive especially if the original lexicon is big. Translating polysemous words can be difficult too, because it is unclear which meaning to use. For example, the word *terrific* (*awesome*, *horrible*) has two meanings with opposite polarities. By using the polarity value from the original lexicon the correct meaning can be guessed, but not in all cases. The word *cool* (*cold*, *awesome*) can have two meanings both with positive polarity but not with the same intensity. Most of the existing lexicons are for general use, so during the translation process we had to consider the domain in which the translated lexicon will be used and in the case of some words the original polarity value should be altered.

2.2 Bootstrapping Sentiment Lexicon

To overcome the above mentioned problems we implemented methods which can automatically create sentiment lexicons. The first method is a corpus-based which exploits a document set which is annotated with polarity labels. The annotation can be done in various ways. The most accurate one is by annotating it by hand. It can be done automatically as well, using an existing sentiment analysis system. For example, this system can be a simple n-gram based model trained on text from another genre. The automatic method can yield a noisy annotation but a large amount of data filters noise. The polarity of a given word can be computed using pointwise mutual information [5]. The method gives a polarity value for all words in the corpus which reflects the positiveness and negativeness of the given word in that domain. Additionally, we scale these values into the $[-5, 5]$ interval. In the following we will refer to lexicons created with this method with the name **pmi**.

2.3 Extending Lexicons

In this section we propose dictionary-based methods for extending seed lexicons. The input seed lexicon contains only a low number of words with their polarity values. The extension is based on similarity measures between words, more precisely we add words to the extended lexicons which are similar to those which are already in the seed lexicon. By using a similarity measure which reflects the aspects of a domain we not only extend the input lexicon, but also adapt it to the given domain.

To assemble the input seed lexicon we created a semi-automated method. We trained an n-gram based sentiment analysis system with maximum entropy classifier on the training portion of the given corpus. Using the trained model it is possible to extract those words that are most likely to occur in positive and negative texts, respectively. From these we used 20 words for both polarity classes in the seed lexicon. Again we scaled the polarity values into the $[-5, 5]$ interval.

WordNet The input of the first extension method is a seed lexicon and a wordnet in a given language. WordNets are large lexical databases which contain words grouped into sets of synonyms (synsets). Synsets are linked by means of conceptual-semantic and lexical relations. Our first similarity measure over words is based on wordnets. Our hypothesis is that the polarity of a word and all of its synonyms is equal. The extension process is as follows. Initially each synset has a polarity value of 0. We iterate over all words in the seed lexicon and assign the actual seed word's polarity value to those synsets in which it appears. Additionally, we used the relation between synsets, namely which sets have similar or opposite meanings. For this we used the *similar_to* and *hyponym* relations in the wordnet. We assign the polarity value or its inverse of a synonym set to all related synsets depending on the relation type. If a synset is related to multiple synsets with non 0 polarity value, we calculate their average. In the last step, we add all the words with the appropriate value to the extended lexicon which are in a synset with a polarity value different than 0. A word type can be in multiple synsets with different polarity values. For example the word *terrific* is included in the following positive and negative synsets {*wonderful*, terrific, fantastic} and {*terrifying*, terrific}, where the seed words are *wonderful* and *terrifying*. In such cases we calculated the average of these polarity values.

The method can be run iteratively, the output of a step can be used as the input of the next one further expanding the lexicon in each step. An important fact is that some words can be added to the extended lexicon with wrong polarity values in an iteration step. For example, in the IT domain if the *silent* word is used as a positive seed word, the *uncommunicative* will be added as positive to the extended lexicon which does not have any polarity in this domain. Because of this, after some iteration step the extended lexicon becomes too noisy. Furthermore, wordnets are general lexical resources, thus the extracted word similarities are not domain dependent. For this reason it is important to start with a seed lexicon which is already domain-specific, this way the extension is aware of the specifics of a given domain. For our experiments we used the Hungarian WordNet [6].

Word Clusters We developed another word similarity measure which is more aware of the specifics of the given domain. For this we used the Brown clustering algorithm [7]. It is a hierarchical clustering of words based on the context in which they occur. The input of this method is a seed lexicon as before and an unlabeled corpus from a given domain. Similar to the previous method, our hypothesis is that the polarities of words in the same cluster are equal. We build clusters on the unlabeled dataset. The initial step of the algorithm is to assign 0 polarity value to all clusters. The next step is to iterate over all words in the seed lexicon and assign the polarity value of the actual seed word to the cluster which contains it. If a cluster contains multiple seed words, we calculate their average value. Lastly, we add words with the appropriate polarity value to the extended lexicon which are in a cluster with not 0 value. The method has one parameter which is the number of clusters to use. If we use a small number of clusters, words which are not similar can be in the same cluster, which causes that the extension assigns wrong polarity values to some words. Inversely, if we use too many clusters, just a small number of new words will be added to the new lexicon. The main advantage of this method in contrast with the wordnet based one is that the clustering algorithm which uses domain-specific texts can capture word similarities which are specific to the given domain. This way it is capable of domain-adapting the input lexicon.

3 Data

In this section we present the used corpora. For our experiments we used two databases: one with texts from news sites and one with IT related product reviews.

The *OpinHuBank* [3] is a corpus created directly for sentiment analysis tasks and contains texts from a general domain using various Hungarian news sites, blogs and forums about sports, politics, economics, etc. Each text instance is an at least 7 token long sentence. The sentences were annotated by 5 annotators with positive, negative or neutral labels. We only used the ones with polarity, more precisely those which were annotated at least by three positive or three negative labels. This way we got 882 positive and 1629 negative sentences in the **opinhu** corpus.

We created a domain specific corpus out of IT product reviews. For this we used the content of a Hungarian site called *árukereső*[1]. This site contains reviews about a wide range of products from which we only used the ones from the PC and electronic products (TV, digital cameras, etc.) categories. The reviewers on this site have to provide pros and cons when writing a review. We used these as positive and negative texts respectfully. Furthermore, we applied filtering on the texts in such a way that we only kept reviews that are one sentence long. The resulting **prodrev** database consists of 3573 positive and 3149 negative sentences.

4 Results

The goal of this work was to create methods to automatically assemble sentiment lexicons which are useful in sentiment analysis tasks. To comparatively evaluate

[1] www.arukereso.hu

lexicons, we defined a sentiment analysis task in which we classify sentences into positive and negative classes and the system was strongly built upon the lexicons. We used a maximum entropy classifier with lemmatized unigrams and lexicon based features. We define the usefulness of a lexicon given a corpus with the accuracy of the classifier system which uses that lexicon. The higher the accuracy, the more useful the lexicon is. In the following we consider a word as *sentiment word* if it is included in the given lexicon and its absolute polarity value is at least 1. A sentiment word is positive or negative depending on the sign of its polarity value. The lexicon based features are the following (an example can be seen in Table 1):

– the sentiment words in the text (in their original form)
– the overall values of positive and negative words respectively
– the overall values of sentiment words
– pairs made of the polarity of a sentiment word and its preceding or following lexical neighbor

Table 1. An example sentence and the features extracted from it. The sentiment word in the sentence is *better*, which has 5.0 polarity value.

Sentence:	The laptop's display has **better** parameters!
Lemmatized unigrams:	the, laptop, display, have, good, parameter, !
sentiment words:	better
Overall values:	POSITIVE=5.0, NEGATIVE=0.0, POLARITY=5.0
Neighbors:	has_POSITIVE, POSITIVE_parameter

Table 2. Extension of seed lexicon with wordnet (wn) and cluster based methods. The accuracies on opinhu and prodrev corpora were measured using 10-fold cross-validation.

opinhu-seed	86.2
opinhu-seed-wn-1	86.4
opinhu-seed-wn-2	85.9
opinhu-seed-wn-3	86.3
opinhu-seed-wn-4	86.0
opinhu-seed-cluster-15	86.7
opinhu-seed-cluster-15-t3	86.8

prodrev-seed	90.7
prodrev-seed-wn-1	90.8
prodrev-seed-wn-2	90.5
prodrev-seed-wn-3	90.8
prodrev-seed-wn-4	90.9
prodrev-seed-cluster-18	90.8
prodrev-seed-cluster-19-t3	90.8

The result of the systems using the lexicon extending techniques can be seen in Table 2. In the case of both the opinhu and prodrev databases, we created a seed lexicon with the semi-automatic method which was presented earlier. The tables show the accuracy of the sentiment analysis systems, which was calculated using 10-fold cross-validation. The notation *wn* indicates the usage of the wordnet based word similarities and the number after that gives the number of iterations we ran. In the case of the opinhu corpus we achieved the highest increase in accuracy with 1 iteration while in the case of prodrev 4 iterations was the best. In both cases, after the 5^{th} iteration the lexicons became too noisy and the results begun to decrease. In the last two rows of the tables, the results of the clustering based extension can be seen. The number at the end of the lines shows the level where the cluster hierarchy was cut and *t3* indicates that we filtered out words from the lexicon which have a frequency of at most 3 in the corpus. This technique was better in the case of the opinhu corpus, and slightly worse in the case of prodrev.

Table 3. Achieved accuracies using different lexicons on opinhu and prodrev.

	opinhu	prodrev
baseline-opinhu	86.1	70.1
baseline-prodrev	61.6	90.0
opinhu-seed-cluster-15-t3	86.8	90.1
prodrev-seed-wn-4	86.2	90.9
translated	88.4	90.2
opinhu-pmi	96.3	90.0
prodrev-pmi	84.3	91.9
prodrev2-pmi	-	91.0

In Table 3, the results of the baseline systems which used only lemmatized unigrams as features can be seen for both corpora, along with the best extended lexicons, the bootstrapped (*pmi*) lexicons and the manually *translated* lexicon (Section 2.1). Two baseline systems had been created, the first was trained on the opinhu corpus and the second on prodrev. The results show that the system not being trained on the same domain as the test corpus resulted in a significantly lower accuracy score. Furthermore, it can be seen that an increase can be achieved with the extending techniques comparing with the baselines if the lexicon is in the appropriate domain. If not, this increase is much smaller. The *translated* lexicon caused 2.3% increase in the opinhu corpus and only 0.2% in the prodrev database, which is less than the effect of the extended lexicons. The reason for this is that opinhu is not domain-specific and the lexicon which was translated was assembled for a similar text genre. The prodrev corpus is IT specific thus needs a lexicon from the same domain.

The bottom 3 rows of Table 3 shows the results for the bootstrapped lexicons. The prefix of each line indicates the annotated corpus which was used to create the lexicon. In those cases where the corpus used for the creation of the lexicon is the same as the corpus on which the sentiment analysis system was evaluated, the results show a theoretical maximum. This maximum shows the accuracy which can be achieved if we have a perfect lexicon for that corpus. It can be seen that these lexicons are not useful for the other domains as they can even decrease the results as well (prodrev-pmi lexicon on the opinhu corpus). We also tried to create a lexicon using texts from the domain of prodrev. For this we created the *prodrev2* corpus, which consists of those positive and negative reviews from the árukereső site that are not one sentence long (shorter and longer). Using this we managed to outperform the lexicons based on the extension methods.

5 Conclusions

In this work we focused on how to create sentiment lexicons automatically, which are useful in sentiment analysis tasks. We presented a technique to create lexicons from scratch by using annotated texts. We also gave methods for extending and adapting lexicons by using two types of word similarity measures. The input of these methods is a small seed lexicon (which we created semi-automatically) and/or (un)labeled domain-specific texts. Our results empirically underpin that it is important to use lexicons which are aware of the specificities of the domain on which the sentiment analysis system operates and by using a lexicon from a different domain the results can even be decreased. Although we achieved an increase in accuracy with the automatically created lexicons on the opinhu corpus, the best results were given by the manually assembled (and translated) lexicon. From this we can conclude that the manually created lexicons are better, but they are much more expensive and it is hard to create one for all domains, thus automatic methods are needed. In the IT specific domain we managed to reduce the errors by 10%. The results show that the proposed automatic methods are useful for increasing the performance of sentiment analysis systems in all domains.

References

1. Baccianella, S., Esuli, A., Sebastiani, F.: SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). (2010)
2. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the conference on human language technology and empirical methods in natural language processing, Association for Computational Linguistics (2005) 347–354
3. Miháltz, M.: OpinHuBank: szabadon hozzáférhető annotált korpusz magyar nyelvű véleményelemzéshez. In: IX. Magyar Számítógépes Nyelvészeti Konferencia. (2013) 343–345
4. Liu, B.: Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies **5**(1) (2012) 1–167
5. Turney, P.D., Littman, M.L.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems (TOIS) **21**(4) (2003) 315–346
6. Miháltz, M., Hatvani, Cs., Kuti, J., Szarvas, Gy., Csirik, J., Prószéky, G., Váradi, T.: Methods and results of the Hungarian WordNet project. In: Proceedings of the Fourth Global WordNet Conference (GWC-2008), Citeseer (2008)
7. Brown, P.F., Desouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. Computational linguistics **18**(4) (1992) 467–479