A Corpus-based Method for Extracting Paraphrases of Emotion Terms

Fazel Keshtkar
University of Ottawa
Ottawa, ON, K1N 6N5, Canada
akeshtka@site.uOttawa.ca

Diana Inkpen
University of Ottawa
Ottawa, ON, K1N 6N5, Canada
diana@site.uOttawa.ca

Abstract

Since paraphrasing is one of the crucial tasks in natural language understanding and generation, this paper introduces a novel technique to extract paraphrases for emotion terms, from non-parallel corpora. We present a bootstrapping technique for identifying paraphrases, starting with a small number of seeds. WordNet Affect emotion words are used as seeds. The bootstrapping approach learns extraction patterns for six classes of emotions. We use annotated blogs and other datasets as texts from which to extract paraphrases, based on the highest-scoring extraction patterns. The results include lexical and morpho-syntactic paraphrases, that we evaluate with human judges.

1 Introduction

Paraphrases are different ways to express the same information. Algorithms to extract and automatically identify paraphrases are of interest from both linguistic and practical points of view. Many major challenges in Natural Language Processing applications, for example multi-document summarization, need to avoid repetitive information from the input documents. In Natural Language Generation, paraphrasing is employed to create more varied and natural text. In our research, we extract paraphrases for emotions, with the goal of using them to automatically-generate emotional texts (such as friendly or hostile texts) for conversations between intelligent agents and characters in educational games. Paraphrasing is applied to generate text with more variety. To our knowledge, most current applications manually collect paraphrases for specific applications, or they use lexical resources such as WordNet (Miller et al., 1993) to identify paraphrases.

This paper introduces a novel method for extracting paraphrases for emotions from texts. We focus on the six basic emotions proposed by Ekman (1992): *happiness*, *sadness*, *anger*, *disgust*, *surprise*, and *fear*.

We describe the construction of the paraphrases extractor. We also propose a *k*-window algorithm for selecting contexts that are used in the paraphrase extraction method. We automatically learn patterns that are able to extract the emotion paraphrases from corpora, starting with a set of seed words. We use data sets such as blogs and other annotated corpora, in which the emotions are marked. We use a large collection of non-parallel corpora which are described in Section 3. These corpora contain many instances of paraphrases different words to express the same emotion.

An example of sentence fragments for one emotion class, *happiness*, is shown in Table 1. From them, the paraphrase pair that our method will extract is:

```
"so happy to see"
"very glad to visit".
```

In the following sections, we give an overview of related work on paraphrasing in Section 2. In Section 3 we describe the datasets used in this work. We explain the details of our paraphrase extraction method in Section 4. We present results of our evaluation and discuss our results in Section 5, and finally in Section 6 we present the conclusions and future work.

Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, pages 35–44,
Los Angeles, California, June 2010.

his little boy was so happy to see him
princess and she were very glad to visit him

Table 1: Two sentence fragments (candidate contexts) from the emotion class *happy*, from the blog corpus.

2 Related Work

Three main approaches for collecting paraphrases were proposed in the literature: manual collection, utilization of existing lexical resources, and corpus-based extraction of expressions that occur in similar contexts (Barzilay and McKeown, 2001). Manually-collected paraphrases were used in natural language generation (NLG) (Iordanskaja et al., 1991). Langkilde et al. (1998) used lexical resources in statistical sentence generation, summarization, and question-answering. Barzilay and McKeown (2001) used a corpus-based method to identify paraphrases from a corpus of multiple English translations of the same source text. Our method is similar to this method, but it extracts paraphrases only for a particular emotion, and it needs only a regular corpus, not a parallel corpus of multiple translations.

Some research has been done in paraphrase extraction for natural language processing and generation for different applications. Das and Smith (2009) presented a approach to decide whether two sentences hold a paraphrase relationship. They applied a generative model that generates a paraphrase of a given sentence, then used probabilistic inference to reason about whether two sentences share the paraphrase relationship. In another research, Wang et. al (2009) studied the problem of extracting technical paraphrases from a parallel software corpus. Their aim was to report duplicate bugs. In their method for paraphrase extraction, they used: sentence selection, global context-based and co-occurrence-based scoring. Also, some studies have been done in paraphrase generation in NLG (Zhao et al., 2009), (Chevelu et al., 2009). Bootstrapping methods have been applied to various natural language applications, for example to word sense disambiguation (Yarowsky, 1995), lexicon construction for information extraction (Riloff and Jones, 1999), and named entity classification (Collins and Singer, 1999). In our research, we use the bootstrapping approach to learn paraphrases for emotions.

3 Data

The text data from which we will extract paraphrases is composed of four concatenated datasets. They contain sentences annotated with the six basic emotions. The number of sentences in each dataset is presented in Table 2. We briefly describe the datasets, as follows.

3.1 LiveJournal blog dataset

We used the blog corpus that Mishne collected for his research (Mishne, 2005). The corpus contains 815,494 blog posts from Livejournal [1], a free weblog service used by millions of people to create weblogs. In Livejournal, users are able to optionally specify their current emotion or mood. To select their emotion/mood users can choose from a list of 132 provided moods. So, the data is annotated by the user who created the blog. We selected only the texts corresponding to the six emotions that we mentioned.

3.2 Text Affect Dataset

This dataset (Strapparava and Mihalcea, 2007) consists of newspaper headlines that were used in the SemEval 2007-Task 14. It includes a development dataset of 250 annotated headlines, and a test dataset of 1000 news headlines. We use all of them. The annotations were made with the six basic emotions on intensity scales of [-100, 100], therefore a threshold is used to choose the main emotion of each sentence.

3.3 Fairy Tales Dataset

This dataset consists in 1580 annotated sentences (Alm et al., 2005), from tales by the Grimm brothers, H.C. Andersen, and B. Potter. The annotations used the extended set of nine basic emotions of Izard (1971). We selected only those marked with the six emotions that we focus on.

3.4 Annotated Blog Dataset

We also used the dataset provided by Aman and Szpakowicz (2007). Emotion-rich sentences were selected from personal blogs, and annotated with the six emotions (as well as a non-emotion class, that we ignore here). They worked with blog posts and collected directly from the Web. First, they prepared

[1] http://www.livejournalinc.com

Dataset	Happiness	Sadness	Anger	Disgust	Surprise	Fear
LiveJournal	7705	1698	4758	1191	1191	3996
TextAffect	334	214	175	28	131	166
Fairy tales	445	264	216	217	113	165
Annotated blog dataset	536	173	115	115	172	179

Table 2: The number of emotion-annotated sentences in each dataset.

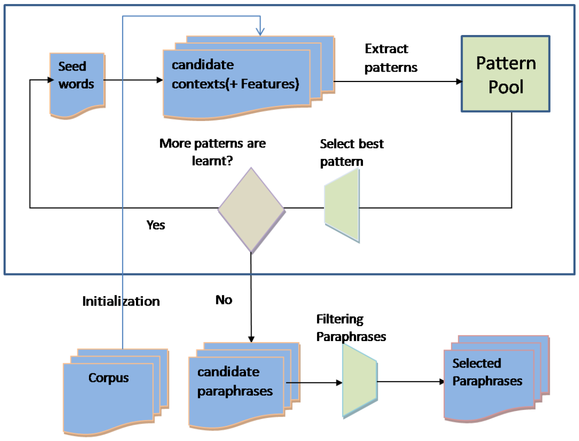

Figure 1: High-level view of the paraphrase extraction method.

a list of seed words for six basic emotion categories proposed by Ekman (1992). Then, they took words commonly used in the context of a particular emotion. Finally, they used the seed words for each category, and retrieved blog posts containing one or more of those words for the annotation process.

4 Method for Paraphrase Extraction

For each of the six emotions, we run our method on the set of sentences marked with the corresponding emotion from the concatenated corpus. We start with a set of seed words form WordNet Affect (Strapparava and Valitutti, 2004), for each emotion of interest. The number of seed words is the following: for happiness 395, for surprise 68, for fear 140, for disgust 50, for anger 250, and for sadness 200. Table 3 shows some of seeds for each category of emotion.

Since sentences are different in our datasets and they are not aligned as parallel sentences as in (Barzilay and McKeown, 2001), our algorithm constructs pairs of similar sentences, based on the local context. On the other hand, we assume that, if the contexts surrounding two seeds look similar, then these contexts are likely to help in extracting new paraphrases.

Figure 1 illustrates the high-level architecture of our paraphrase extraction method. The input to the method is a text corpus for a emotion category and a manually defined list of seed words. Before bootstrapping starts, we run the k-window algorithm on every sentence in the corpus, in order to construct *candidate contexts*. In Section 4.5 we explain how the bootstrapping algorithm processes and selects the paraphrases based on strong surrounding contexts. As it is shown in Figure 1, our method has several stages: extracting candidate contexts, using them to extract patterns, selecting the best patterns, extracting potential paraphrases, and filtering them to obtain the final paraphrases.

4.1 Preprocessing

During preprocessing, HTML and XML tags are eliminated from the blogs data and other datasets, then the text is tokenized and annotated with part of speech tags. We use the Stanford part-of-speech tagger and chunker (Toutanova et al., 2003) to identify noun and verb phrases in the sentences. In the next step, we use a sliding window based on the k-window approach, to identify candidate contexts that contain the target seeds.

4.2 The k-window Algorithm

We use the k-window algorithm introduced by Bostad (2003) in order to identify all the tokens surrounding a specific term in a window with size of $\pm k$. Here, we use this approach to extract candidate patterns for each seed, from the sentences. We start with one seed and truncate all contexts around the seed within a window of $\pm k$ words before and $\pm k$ words after the seed, until all the seeds are processed. For these experiments, we set the value of k to ± 5. Therefore

Happiness: avidness, glad, warmheartedness, exalt, enjoy, comforting, joviality, amorous, joyful, like, cheer, adoring, fascinating, happy, impress, great, satisfaction, cheerful, charmed, romantic, joy, pleased, inspire, good, fulfill, gladness, merry
Sadness: poor, sorry, woeful, guilty, miserable, glooming, bad, grim, tearful, glum, mourning, joyless, sadness, blue, rueful, hamed, regret, hapless, regretful, dismay, dismal, misery, godforsaken, oppression, harass, dark, sadly, attrition
Anger: belligerence, envious, aggravate, resentful, abominate, murderously, greedy, hatred, disdain, envy, annoy, mad, jealousy, huffiness, sore, anger, harass, bother, enraged, hateful, irritating, hostile, outrage, devil, irritate, angry
Disgust: nauseous, sicken, foul, disgust, nausea, revolt, hideous, horror, detestable, wicked, repel, offensive, repulse, yucky, repulsive, queasy, obscene, noisome
Surprise: wondrous, amaze, gravel, marvel, fantastic, wonderful, surprising, marvelous, wonderment, astonish, wonder, admiration, terrific, dumfounded, trounce
Fear: fearful, apprehensively, anxiously, presage, horrified, hysterical, timidity, horrible, timid, fright, hesitance, affright, trepid, horrific, unassertive, apprehensiveness, hideous, scarey, cruel, panic, scared, terror, awful, dire, fear, dread, crawl, anxious, distrust, diffidence

Table 3: Some of the seeds from WordNet Affect for each category of emotion.

Candidate context: He was further annoyed by the jay bird
'PRP VBD RB VBN IN DT NN NN',65,8,'VBD RB',?,was, ?,?,?,He/PRP,was/VBD,further/RB,annoyed,by/IN,the/DT, jay/NN,bird/NN,?,?,jay,?,'IN DT NN',2,2,0,1

Table 4: An example of extracted features.

the longest candidate contexts will have the form $w_1, w_2, w_3, w_4, w_5, seed, w_6, w_7, w_8, w_9, w_{10}, w_{11}$. In the next subsection, we explain what features we extract from each candidate context, to allow us to determine similar contexts.

4.3 Feature Extraction

Previous research on word sense disambiguation on contextual analysis has acknowledged several local and topical features as good indicators of word properties. These include surrounding words and their part of speech tags, collocations, keywords in contexts (Mihalcea, 2004). Also recently, other features have been proposed: bigrams, named entities, syntactic features, and semantic relations with other words in the context.

We transfer the candidate phrases extracted by the sliding *k*-window into the vector space of features. We consider features that include both lexical and syntactic descriptions of the paraphrases for all pairs of two candidates. The lexical features include the sequence of tokens for each phrase in the paraphrase pair; the syntactic feature consists of a sequence of part-of-speech (PoS) tags where equal words and words with the same root and PoS are marked. For example, the value of the syntactic feature for the pair ``so glad to see'' and ``very happy to visit'' is “$RB_1\ JJ_1\ TO\ VB_1$” and ”$RB_1\ JJ_2\ TO\ VB_2$”, where indices indicate word equalities. However, based on the above evidences and our previous research, we also investigate other features that are well suited for our goal. Table 5 lists the features that we used for paraphrase extraction. They include some term frequency features. As an example, in Table 4 we show extracted features from a relevant context.

4.4 Extracting Patterns

From each candidate context, we extracted the features as described above. Then we learn extraction patterns, in which some words might be substituted by their part-of-speech. We use the seeds to build initial patterns. Two candidate contexts that contain the same seed create one positive example. By using each initial seed, we can extract all contexts surrounding these positive examples. Then we select the stronger ones. We used Collins and Singer method (Collins and Singer, 1999) to compute the strength of each example. If we consider x as a context, the strength as a positive example of x is de-

Features	Description
F1	Sequence of part-of-speech
F2	Length of sequence in bytes
F3	Number of tokens
F4	Sequence of PoS between the seed and the first verb before the seed
F5	Sequence of PoS between the seed and the first noun before the seed
F6	First verb before the seed
F7	First noun before the seed
F8	Token before the seed
F9	Seed
F10	Token after the seed
F11	First verb after the seed
F12	First noun after the seed
F13	Sequence of PoS between the seed and the first verb after the seed
F14	Sequence of PoS between the seed and the first noun after the seed
F15	Number of verbs in the candidate context
F16	Number of nouns in the candidate context
F17	Number of adjective in the candidate context
F18	Number of adverbs in the candidate context

Table 5: The features that we used for paraphrase extraction.

fined as:

$$Strength(x) = count(x+)/count(x) \qquad (1)$$

In Equation 1, $count(x+)$ is the number of times context x surrounded a seed in a positive example and $count(x)$ is frequency of the context x. This allows us to score the potential pattern.

4.5 Bootstrapping Algorithm for Paraphrase Extraction

Our bootstrapping algorithm is summarized in Figure 2. It starts with a set of seeds, which are considered initial paraphrases. A set of extraction patterns is initially empty. The algorithm generates candidate contexts, from the aligned similar contexts. The candidate patterns are scored by how many paraphrases they can extract. Those with the highest scores are added to the set of extraction patterns. Using the extended set of extraction patterns, more paraphrase pairs are extracted and added to the set of paraphrases. Using the enlarged set of paraphrases, more extraction patterns are extracted. The process keeps iterating until no new patterns or no new paraphrases are learned.

Our method is able to accumulate a large lexicon of emotion phrases by bootstrapping from the manually initialized list of seed words. In each iteration, the paraphrase set is expanded with related phrases found in the corpus, which are filtered by using a measure of strong surrounding context similarity. The bootstrapping process starts by selecting a subset of the extraction patterns that aim to extract the paraphrases. We call this set the pattern pool. The phrases extracted by these patterns become candidate paraphrases. They are filtered based on how many patterns select them, in order to produce the final paraphrases from the set of candidate paraphrases.

5 Results and Evaluation

The result of our algorithm is a set of extraction patterns and a set of pairs of paraphrases. Some of the paraphrases extracted by our system are shown in Table 6. The paraphrases that are considered correct are shown under *Correct paraphrases*. As explained in the next section, two human judges agreed that these are acceptable paraphrases. The results considered incorrect by the two judges are shown un-

Algorithm 1: Bootstrapping Algorithm.

```
For each seed for an emotion
  Loop until no more paraphrases or no more contexts are learned.
    1- Locate the seeds in each sentence
    2- Find similar contexts surrounding a pair of two seeds
    3- Analyze all contexts surrounding the two seeds to extract
       the strongest patterns
    4- Use the new patterns to learn more paraphrases
```

Figure 2: Our bootstrapping algorithm for extracting paraphrases.

der *Incorrect paraphrases*. Our algorithm learnt 196 extraction patterns and produced 5926 pairs of paraphrases. Table 7 shows the number of extraction patterns and the number of paraphrase pairs that were produced by our algorithm for each class of emotions. For evaluation of our algorithm, we use two techniques. One uses human judges to judge if a sample of paraphrases extracted by our method are correct; we also measures the agreement between the judges (See Section 5.1). The second estimates the recall and the precision of our method (See Section 5.2. In the following subsections we describe these evaluations.

5.1 Evaluating Correctness with Human Judges

We evaluate the correctness of the extracted paraphrase pairs, using the same method as Brazilay and McKeown (2001). We randomly selected 600 paraphrase pairs from the lexical paraphrases produced by our algorithm: for each class of emotion we selected 100 paraphrase pairs. We evaluated their correctness with two human judges. They judged whether the two expressions are good paraphrases or not.

We provided a page of guidelines for the judges. We defined paraphrase as ”approximate conceptual equivalence”, the same definition used in (Barzilay and McKeown, 2001). Each human judge had to choose a ”Yes” or ”No” answer for each pair of paraphrases under test. We did not include example sentences containing these paraphrases. A similar Machine Translation evaluation task for word-to-word translation was done in (Melamed, 2001).

Figure 3 presents the results of the evaluation: the correctness for each class of emotion according to judge A, and according to judge B. The judges were graduate students in computational linguistics, native speakers of English.

We also measured the agreement between the two judges and the Kappa coefficient (Siegel and Castellan, 1988). If there is complete agreement between two judges $Kappa$ is 1, and if there is no agreement between the judges then $Kappa = 0$. The $Kappa$ values and the agreement values for our judges are presented in Figure 4.

The inter-judge agreement over all the paraphrases for the six classes of emotions is 81.72%, which is 490 out of the 600 paraphrases pairs in our sample. Note that they agreed that some pairs are good paraphrases, or they agreed that some pairs are not good paraphrases, that is why the numbers in Figure 4 are higher than the correctness numbers from Figure 3. The $Kappa$ coefficient compensates for the chance agreement. The $Kappa$ value over all the paraphrase pairs is 74.41% which shows a significant agreement.

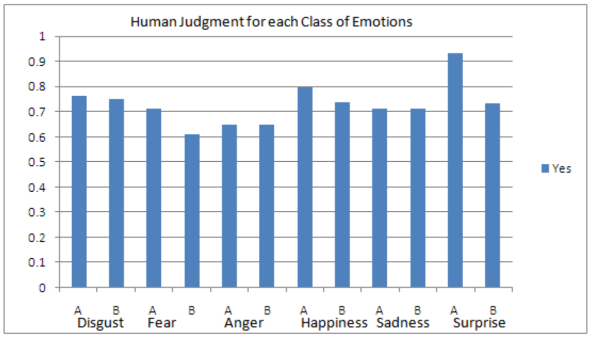

Figure 3: The correctness results according the judge A and judge B, for each class of emotion.

5.2 Estimating Recall

Evaluating the *Recall* of our algorithm is difficult due to following reasons. Our algorithm is not able to cover all the English words; it can only detect

Disgust
Correct paraphrases: being a wicked::getting of evil; been rather sick::feeling rather nauseated; feels somewhat queasy::felt kind of sick; damn being sick::am getting sick
Incorrect paraphrases: disgusting and vile::appealing and nauseated; get so sick::some truly disgusting
Fear

Correct paraphrases:
was freaking scared::was quite frightened; just very afraid::just so scared;
tears of fright::full of terror; freaking scary::intense fear;

Incorrect paraphrases: serious panic attack::easily scared; not necessarily fear::despite your fear
Anger
Correct paraphrases: upset and angry::angry and pissed; am royally pissed::feeling pretty angry; made me mad::see me angry; do to torment::just to spite
Incorrect paraphrases: very pretty annoying::very very angry; bitter and spite::tired and angry
Happiness
Correct paraphrases: the love of::the joy of; in great mood::in good condition; the joy of::the glad of; good feeling::good mood
Incorrect paraphrases: as much eagerness::as many gladness; feeling smart::feel happy
Sadness
Correct paraphrases: too depressing::so sad; quite miserable::quite sorrowful; strangely unhappy::so misery; been really down::feel really sad
Incorrect paraphrases: out of pity::out of misery; akward and depressing::terrible and gloomy
Surprise
Correct paraphrases: amazement at::surprised by; always wonder::always surprised; still astounded::still amazed; unexpected surprise::got shocked
Incorrect paraphrases: passion and tremendous::serious and amazing; tremendous stress::huge shock

Table 6: Examples of paraphrases extracted by our algorithm (correctly and incorrectly).

Class of Emotion	# Paraphrases Pairs	# Extraction Patterns
Disgust	1125	12
Fear	1004	31
Anger	670	47
Happiness	1095	68
Sadness	1308	25
Surprise	724	13
Total	5926	196

Table 7: The number of lexical and extraction patterns produced by the algorithm.

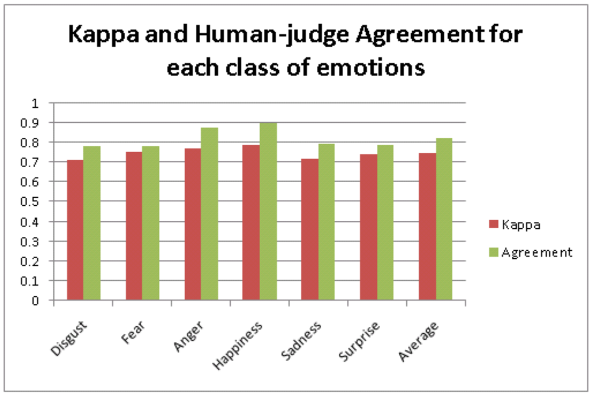

Figure 4: The $Kappa$ coefficients and the agreement between the two human judges.

paraphrasing relations with words which appeared in our corpus. Moreover, to compare directly with an electronic thesaurus such as WordNet is not feasible, because WordNet contains mostly synonym sets between words, and only a few multi-word expressions. We decided to estimate recall manually, by asking a human judge to extract paraphrases by hand from a sample of text. We randomly selected 60 texts (10 for each emotion class) and asked the judge to extract paraphrases from these sentences. For each emotion class, the judge extracted expressions that reflect the emotion, and then made pairs that were conceptually equivalent. It was not feasible to ask a second judge to do the same task, because the process is time-consuming and tedious.

In Information Retrieval, *Precision* and *Recall* are defined in terms of a set of retrieved documents and a set of relevant documents [2]. In the following sections we describe how we compute the *Precision* and *Recall* for our algorithm compared to the manually extracted paraphrases.

Category of Emotions	Precision	Recall
Disgust	82.33%	92.91%
Fear	82.64%	88.20%
Anger	93.67%	80.57%
Happiness	82.00%	90.89%
Sadness	82.00%	89.88%
Surprise	79.78%	89.50%
Average	84.23%	88.66%

Table 8: Precision and Recall for a sample of texts, for each category of emotion, and their average.

From the paraphrases that were extracted by the algorithm from the same texts, we counted how many of them were also extracted by the human judge. Equation 2 defines the Precision. On average, from 89 paraphrases extracted by the algorithm, 74 were identified as paraphrases by the human judge (84.23%). See Table 8 for the values for all the classes.

$$P = \frac{\#\ \textit{Correctly Retrieved Paraphrases by the Algorithm}}{\textit{All Paraphrases Retrieved by the Algorithm}} \qquad (2)$$

For computing the *Recall* we count how many of the paraphrases extracted by the human judge were correctly extracted by the algorithm (Equation 3).

$$R = \frac{\#\ \textit{Correctly Retrieved Paraphrases by the Algorithm}}{\textit{All Paraphrases Retrieved by the Human Judge}} \qquad (3)$$

5.3 Discussion and Comparison to Related Work

To the best of our knowledge, no similar research has been done in extracting paraphrases for emotion terms from corpora. However, Barzilay and McKeown (2001) did similar work to corpus-based iden-

[2] http://en.wikipedia.org/wiki/

tification of general paraphrases from multiple English translations of the same source text. We can compare the pros and cons of our method compared to their method. The advantages are:

- In our method, there is no requirement for the corpus to be parallel. Our algorithm uses the entire corpus together to construct its bootstrapping method, while in (Barzilay and McKeown, 2001) the parallel corpus is needed in order detect positive contexts.

- Since we construct the candidate contexts based on the *k*-window approach, there is no need for sentences to be aligned in our method. In (Barzilay and McKeown, 2001) sentence alignment is essential in order to recognize identical words and positive contexts.

- The algorithm in (Barzilay and McKeown, 2001) has to find positive contexts first, then it looks for appropriate patterns to extract paraphrases. Therefore, if identical words do not occur in the aligned sentences, the algorithm fails to find positive contexts. But, our algorithm starts with given seeds that allow us to detect positive context with the *k*-window method.

A limitation of our method is the need for the initial seed words. However, obtaining these seed words is not a problem nowadays. They can be found in on line dictionaries, WordNet, and other lexical recourses.

6 Conclusion and Future Work

In this paper, we introduced a method for corpus-based extraction of paraphrases for emotion terms. We showed a method that used a bootstrapping technique based on contextual and lexical features and is able to successfully extract paraphrases using a non-parallel corpus. We showed that a bootstrapping algorithm based on contextual surrounding context features of paraphrases achieves significant performance on our data set.

In future work, we will extend this techniques to extract paraphrases from more corpora and for more types of emotions. In terms of evaluation, we will use the extracted paraphrases as features in machine learning classifiers that classify candidate sentences into classes of emotions. If the results of the classification are good, this mean the extracted paraphrases are of good quality.

References

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the Human Language Technology Conference Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*.

Saima Aman and Stan Szpakowicz. 2007. Identifying expressions of emotion in text. In *TSD*, pages 196–205.

Regina Barzilay and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceeding of ACL/EACL, 2001, Toulouse*.

Thorstein Bostad. 2003. *Sentence Based Automatic Sentiment Classification*. Ph.D. thesis, University of Cambridge, Computer Speech Text and Internet Technologies (CSTIT), Computer Laboratory, Jan.

Jonathan Chevelu, Thomas Lavergne, Yves Lepage, and Thierry Moudenc. 2009. Introduction of a new paraphrase generation tool based on Monte-Carlo sampling. In *Proceedings of ACL-IJCNLP 2009, Singapore*, pages 249–25.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Dipanjan Das and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of ACL-IJCNLP 2009, Singapore*, pages 468–476.

Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion*, 6:169–200.

L. Iordanskaja, Richard Kittredget, and Alain Polguere, 1991. *Natural Language Generation in Artificial Intelligence and Computational Linguistics*. Kluwer Academic.

Carroll E. Izard. 1971. *The Face of Emotion*. Appleton-Century-Crofts., New York.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *COLING-ACL*.

Ilya Dan Melamed. 2001. *Empirical Methods for Exploiting Parallel Texts*. MIT Press.

Rada Mihalcea. 2004. Co-training and self-training for word sense disambiguation. In *Natural Language Learning (CoNLL 2004)*, Boston, May.

George Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller, 1993. *Introduction to Wordnet: An On-Line Lexical Database*. Cognitive Science Laboratory, Princeton University, August.

Gilad Mishne. 2005. Experiments with mood classification in blog posts. *ACM SIGIR*.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, page 10441049. The AAAI Press/MIT Press.

Sidney Siegel and John Castellan, 1988. *Non Parametric Statistics for Behavioral Sciences*. . McGraw-Hill.

Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on the Semantic Evaluations (SemEval 2007), Prague, Czech Republic, June 2007*.

Carlo Strapparava and Alessandro Valitutti. 2004. Wordnet-affect: an affective extension of wordnet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, May 2004*, pages 1083–1086.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL*, pages 252–259.

Xiaoyin Wang, David Lo, Jing Jiang, Lu Zhang, and Hong Mei. 2009. Extracting paraphrases of technical terms from noisy parallel software corpora. In *Proceedings of ACL-IJCNLP 2009, Singapore*, pages 197–200.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196.

Shiqi Zhao, Xiang Lan, Ting Liu, , and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of ACL-IJCNLP 2009, Singapore*, pages 834–842.