Structural Alignment for Comparison Detection

Wiltrud Kessler and **Jonas Kuhn**
Institute for Natural Language Processing
University of Stuttgart
wiltrud.kessler@ims.uni-stuttgart.de

Abstract

There tends to be a substantial proportion of reviews that include explicit textual comparisons between the reviewed item and another product. To the extent that such comparisons can be captured reliably by automatic means, they can provide an extremely helpful input to support a process of choice. As the small amount of available training data limits the development of robust systems to automatically detect comparisons, this paper investigates how to use semi-supervised strategies to expand a small set of labeled sentences. Specifically, we use structural alignment, a method that starts out from a seed set of manually annotated data and finds similar unlabeled sentences to which the labels can be projected. We present several adaptations of the method to our task of comparison detection and show that adding the found expansion sentences slightly improves over a non-expanded baseline in low-resource settings, i.e., when a very small amount of training data is available.

1 Introduction

Sentiment analysis is an NLP task that has received considerable attention in recent years. If we consider the actual situations in which people are interested in aggregated subjective assessments of some product (or location, service etc.) by other users, a typical scenario is that they are in the process of making some choice – such as a purchase decision among a set of candidate products. It is clear that for this decision a plain polarity scoring for entire review texts is of limited use and we need a more detailed analysis. In this work, we focus on what is presumably the most useful kind of expression when it comes to supporting a process of choice: there tends to be a substantial proportion of reviews (about 10% of sentences) that include explicit textual comparisons, e.g., *"X is better than Y"*. To the extent that such subjective comparisons can be captured reliably by automatic means, they can provide an extremely helpful basis for coming up with a decision.

The analysis of comparisons has the disadvantage that data for supervised training can no longer be derived from star ratings. Existing manually annotated sentiment analysis data sets include some proportion of comparisons, however, for a reliable supervised training, a larger data set is required. Moreover, vocabulary differences across product categories make it advisable to use domain-specific training data.

If enough (human and/or financial) resources are available, the most effective approach is of course to invest in quality-controlled manual annotation of a relatively large amount of training data. However, since the higher-level semantic structure of comparisons as they appear in reviews is clear-cut, the problem setting could respond favorably to weakly supervised training strategies that start out from a seed set of manually annotated data. The experiments we present in this paper are exploring this very question.

Comparisons can be mapped to a predicate-argument structure, so we cast the task of detecting them as a semantic role-labeling (SRL) problem (Hou and Li, 2008; Kessler and Kuhn, 2013). Starting with a small set of labeled seed sentences, we use structural alignment (Fürstenau and Lapata, 2009), which has been successfully applied to SRL, to automatically find and annotate sentences that are similar to these seed sentences as a way to get more training data.

There are several challenges that make our task different from a typical SRL setting: Our data is not news, but user-generated data (product reviews), which is much more noisy. We have a

Proceedings of Recent Advances in Natural Language Processing, pages 275–281,
Hissar, Bulgaria, Sep 7–9 2015.

smaller, more fixed set of roles for the arguments (two entities that are compared in some aspect), but these arguments are further away from the predicates. And, like all sentiment-related task, we have to deal with subjectivity.

In this work we want to investigate whether structural alignment can successfully be used for getting additional training data for the task of comparison detection. We present some adaptations of the method to our task of comparison detection and experiment with varying numbers of seed sentences and gathered expansion sentences.

2 Related work

Sentiment analysis has in recent years moved from the document-level prediction of polarity or star rating to a more fine-grained analysis. Jindal and Liu (2006a) are the first to specifically distinguish comparison sentences from other sentences in product reviews. In follow-up work, Jindal and Liu (2006b) detect comparison arguments with label sequential rules and Ganapathibhotla and Liu (2008) identify the preferred entity in a ranked comparison. Xu et al. (2011) use Conditional Random Fields in relation extraction approach. We follow previous work (Hou and Li, 2008; Kessler and Kuhn, 2013) and tackle comparisons with a SRL approach, but move from a completely supervised setting to a semi-supervised one.

Several unsupervised or weakly supervised approaches have been presented for SRL. Gildea and Jurafsky (2002) – the first work that tackles SRL as an independent task – use bootstrapping, where an initial system is trained on the available data, applied to a large unlabeled corpus, and the resulting annotations are then used to re-train the model. Abend et al. (2009) do unsupervised argument identification by using pointwise mutual information to determine which constituents are the most probable arguments. Other approaches use the extensive resources that exist for SRL as a basis, e.g., Swier and Stevenson (2005) leverage VerbNet which lists possible argument structures allowable for each predicate. For comparison detection we do not have extensive resources to tap into. We do however think that a small seed set of comparison sentences can be annotated in reasonable time for any new domain or language. This set may not be sufficiently large for bootstrapping, but it can be used as an initial seed set. In this work, we use structural alignment (Fürstenau and Lapata, 2009) to expand this seed set with similar sentences in a semi-supervised way.

3 Approach

The goal of our work is to get more training data for comparison detection in a semi-supervised way. We implement structural alignment proposed by Fürstenau and Lapata (2009) and Fürstenau and Lapata (2012), a method for finding unlabeled sentences that are similar to existing labeled seed sentences (originally proposed for SRL). The basic hypothesis is that predicates that appear in a similar syntactic and semantic context will behave similarly with respect to their arguments so that the labels from the seed sentences can be projected to the unlabeled sentences. These newly labeled sentences can then be used as additional training data.

3.1 Outline of structural alignment

Given a small set of labeled sentences (seed corpus) and a large set of unlabeled sentences (expansion corpus). We collect expansion sentences for a predicate p of a seed sentence s with the follwing steps for every unlabeled sentence u.

1. *Sentence selection:* Consider u iff it contains a predicate compatible with p.
2. *Argument candidate creation:* Get all argument candidates from s and from u.
3. *Alignment scoring:* Score every possible alignment between the two argument candidate sets.
4. Store best-scoring alignment and its score iff at least one role-bearing node is covered.

When all unlabeled sentences have been processed, we choose the k sentences with the highest alignment similarity scores as expansion sentences for the seed predicate p. We project the labels of the arguments in the seed sentence onto their aligned words in these unlabeled sentences and add the newly labeled sentences to our data.

In the following we will discuss the main steps of the expansion algorithm and give some details. Figure 1 illustrates each step for a pair of example sentences from our data.

3.2 Sentence selection

We consider all sentences with the exact same lemma for the predicate as possible expansion sentences. In contrast to the original approach, we use the part of speech (POS) tag instead of the lemma

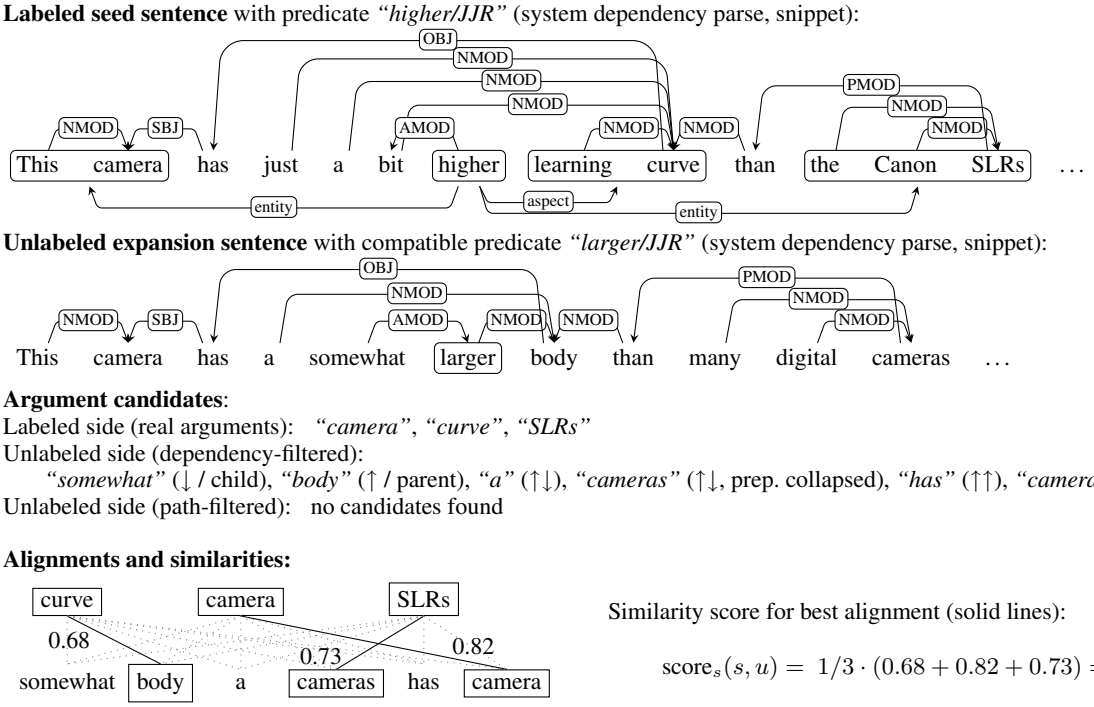


Figure 1: Steps of structural alignment for an example seed and an example unlabeled sentence.

for all adjectives and adverbs in comparative or superlative form (see Figure 1 where both predicates are *"JJR"*), as exchanging them is without any influence on the syntactic structure or the arguments of the comparison. Like the original approach, we only consider single-word predicates.

3.3 Argument candidate creation

Fürstenau and Lapata (2009) use the direct descendants and siblings of the predicate as argument candidates (both SRL arguments and non-arguments). In our labeled data, this find only 17% of the actual labeled comparison arguments.

The challenge is to enlarge the set of argument candidates, while keeping the number of candidates manageable so that alignments can be calculated in reasonable time. Similar to what has been proposed for SRL arguments (Xue and Palmer, 2004), we use all ancestors of the predicate until the root and their direct descendants, plus all descendants of the predicate itself. We remove prepositions (Fürstenau and Lapata, 2009) and conjunctions (Fürstenau and Lapata, 2012) which can never be arguments, and add their direct children to the candidate set. We also impose a distance limit and exclude numbers and punctuation. Applied to our labeled data, this *dependency-filtered* method finds 87% of all real arguments.

As a second method (*path-filtered*), we get the paths from the predicate to each argument in the labeled sentence and search for the exact same paths (compared by dependency relations) in the unlabeled sentence. All nodes on the path are extracted as candidates (Fürstenau and Lapata, 2012). The method is very precise, but also often fails to find any candidates.

On labeled side, we only take the actual labeled arguments of the comparison, as our candidate sets are relatively big and noisy and our interest is solely in finding good alignments for the projection of the real arguments. You can see the resulting candidates for the example in Figure 1.

3.4 Alignment scoring

The similarity of an alignment between two sentences s and u is the averaged sum of all word alignment similarities, themselves the averaged sum of different word similarity measures:

$$\text{score}_s(s,u) = \frac{1}{|M|}\sum_{i=1}^{|M|}\frac{1}{|S|}\sum_{j\in S}\text{sim}_j(w_i,\sigma(w_i))$$

where M is the set of candidates on labeled side, $w_i \in M$ one of these candidates, $\sigma(w_i)$ the candidate on unlabeled side aligned with w_i, and S is the set of similarities to calculate. Unaligned w_i receive a word similarity of zero.

name	value w_i	value $\sigma(w_i)$	$\text{sim}_j(\cdot)$	explanation
sim_{vs}	$\vec{v}$(slrs)	$\vec{v}$(cameras)	0.91	Cosine similarity of co-occurrence vectors $\vec{v}()$
$\text{sim}_{\text{neigh}}$	$\vec{v}$(canon), $\vec{v}$(i)	$\vec{v}$(digital), $\vec{v}$(but)	0.78	(sim_{vs} of left neighbors + sim_{vs} of right neighbors) / 2
sim_{dep}	PMOD, NNP	PMOD, NNS	0.75	Dependency relation similarity (0.5 same, 0 else) + POS sim. (0.5 same, 0.25 same universal POS, 0 else)
sim_{tok}	6	5	0.50	Similarity of distance (# tokens) of candidate from predicate $1/(\lvert d_{\text{tok}}(w_i, p) - d_{\text{tok}}(\sigma(w_i), \sigma(p))\rvert + 1)$.
sim_{lev}	$\uparrow 2 \downarrow 2$	$\uparrow 1 \downarrow 2$	0.75	Similarity in number of "up"s ($\uparrow$) and "down"s ($\downarrow$) on the dependency path from argument to predicate. The $\uparrow$ and $\downarrow$ parts are calculated separately and averaged.
sim_{path}	$\uparrow$ bit $\downarrow$ curve, than	$\downarrow$ body, than	0.70	Average sim_{dep} of all words on the the dependency path from argument to the predicate. The $\uparrow$ and $\downarrow$ parts are calculated separately, similarity for unpaired words is 0.

Table 1: Similarity measures for word alignment similarity. Columns 2–4 give the compared values and similarities for the example from Figure 1 with *"SLRs"* as w_i and *"cameras"* as $\sigma(w_i)$.

We compare the syntactic and semantic similarity of the two candidates with a variety of similarity measures that are listed in Table 1 along with values they take for the example from Figure 1.

We use two combinations of similarity measures: *flat* similarities only ($S = \{\text{vs}, \text{dep}\}$) which corresponds to the similarity measures used in the original work, and similarities that include *context* (all, $S = \{\text{vs}, \text{neigh}, \text{dep}, \text{tok}, \text{lev}, \text{path}\}$).

4 Experiments

4.1 Data

As our core labeled data set we use comparison sentences from English camera reviews[1] (Kessler and Kuhn, 2014). We divide the data into five folds and use one fold as seed data and the rest as test data. The full **seed data** contains 342 sentences with 415 predicates. The **test data** contains 1365 sentences with 1693 predicates.

As the unlabeled **expansion data**, we use a set of 280.000 camera review sentences from `epinions.com`. Note that expansion sentences are never used in testing, we always only test on human-annotated data.

To calculate vector space similarities we use co-occurrence vectors (symmetric window of 2 words, retain 2000 most frequent dimensions) extracted from a large set of reviews with a total of 40 million tokens. This set includes the above expansion corpus, the electronics part of the HUGE corpus (Jindal and Liu, 2008) and camera reviews from `amazon.com`.

[1] `http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/reviewcomparisons/`

4.2 System for comparison detection

We retrain the MATE Semantic Role Labeling system (Björkelund et al., 2009)[2] on our data and use a typical pipeline setting with three classification steps: predicate identification, argument identification and argument classification. We distinguish three argument types: two entities and one aspect. We use standard SRL features (Johansson and Nugues, 2007) based on the output of the MATE dependency parser. This setup is equivalent to (Kessler and Kuhn, 2013).

4.3 Experimental setup

To evaluate whether the found expansion sentences are useful, we add the k best expansion sentences per seed predicate to the seed data and train on this expanded corpus. We use the test data for evaluation and compare classification performance of training on the expanded seed data with the **baseline** trained on the seed data only.

We test four versions of the expansion:

PATH-FLAT *path-filtered* candidate creation and *flat* similarities (closest to the original work).

DEP-FLAT *dependency-filtered* candidate creation and *flat* similarities.

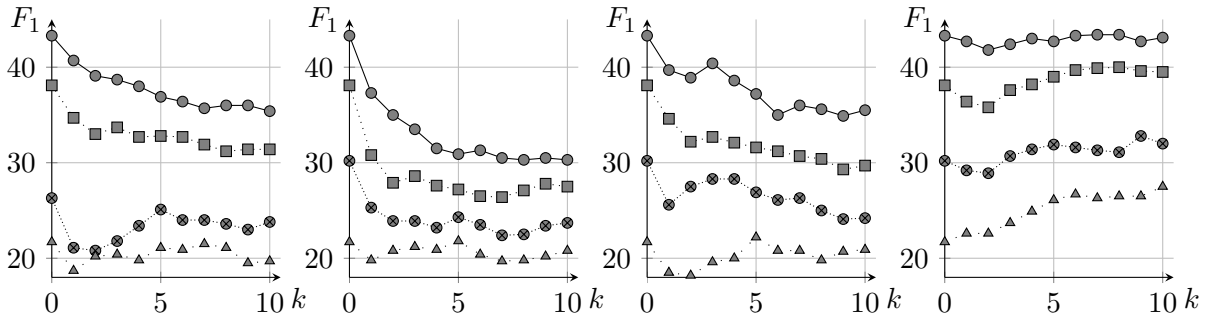
PATH-CONTEXT *path-filtered* candidate creation and *context* similarities.

DEP-CONTEXT *dependency-filtered* candidate creation and *context* similarities.

There are two main questions we investigate:

1. How many seed sentences should be used (varying d)?
2. How many expansion sentences should be used per seed (varying k)?

[2] `http://code.google.com/p/mate-tools/`



(a) PATH-FLAT (b) DEP-FLAT (c) PATH-CONTEXT (d) DEP-CONTEXT

Figure 2: F_1 score for argument identification when using different percentages d of the corpus as seed data (top to bottom: 100%, 50%, 25%, 10%) and expanding with different k numbers of candidates.

We expect that the training data expansion is helpful in low-resource and high-precision settings (i.e., d and k are small). This corresponds to a scenario where only a limited amount of sentences has been annotated for a new domain or language. We consider this to be a more realistic scenario for our task than the one used in (Fürstenau and Lapata, 2012), where a fixed number of training examples per frame is used, as in contrast to SRL we do not expect to know predicates or frames for comparisons in advance.

4.4 Results

Figure 2 shows some results for comparison argument identification in terms of F_1 score. The different curves represent expanding and training on different percentages d of the seed set, from 10% to 100% (full seed set). Note that the lowest setting uses only 34 seed sentences.

The x-axis shows k, the number of expansion sentences added per seed sentence. The value 0 corresponds to the baseline, i.e., training on the seed sentences only. In line with the results reported for SRL, for most cases as k gets larger, the amount of introduced noise outweighs the benefits of additional training data, so performance drops.

For PATH-FLAT, DEP-FLAT and PATH-CONTEXT, almost no setting manages to improve over the non-expanded baseline, every added expansion sentence only decreases performance. For DEP-CONTEXT, in some cases, especially for low values for d there is a small improvement. To illustrate the different sentences selected by the systems, consider this example:

(1) a. *"I felt **more** [comfortable]*$_{\text{aspect}}$ *with [XTi]*$_{\text{entity}}$*"*

b. *"I bought this because my wife didn't feel [comfortable]*$_{\text{aspect}}$ *with all the features/functions of the **more** complex [C5050Z]*$_{\text{entity}}$*."*

c. *"I was much **more** [comfortable]*$_{\text{aspect}}$ *with the [DSC-S75]*$_{\text{entity}}$*"*

Sentence 1a is the seed sentence, sentence 1b is the sentence selected by DEP-FLAT, sentence 1c is selected by DEP-CONTEXT. While choosing *"comfortable"* in sentence 1b to be aligned with the labeled aspect seems like a perfect match in isolation, 1c is a much better choice in context.

Figure 3 shows learning curves for argument identification for each system with the best setting for k (usually 1, 10 for DEP-CONTEXT). All systems except DEP-CONTEXT are nearly always below the baseline. The best value of k for DEP-CONTEXT in our experiments is 10, which is shown in the graph. The results are very similar for all $k \geq 5$, for lower values of k, the results drop below the baseline. The best setting manages to improve over the non-expanded baseline in low resource settings, but the curves get closer to each other when more seed data is added and the effect disappears at the end.

Due to space restrictions we are only able to show argument identification results, but the trends are very similar for predicate identification and argument classification.

4.5 Discussion

If we look at the sentences found by the expansion systems, we can identify two main problems with the extracted sentences.

One problem that affects all sentiment-related tasks is subjectivity. Often sentiment words (or in our case comparison words) appear in non-sentiment (non-comparative) contexts, but these contexts are very hard to distinguish from each other. Consider this example:

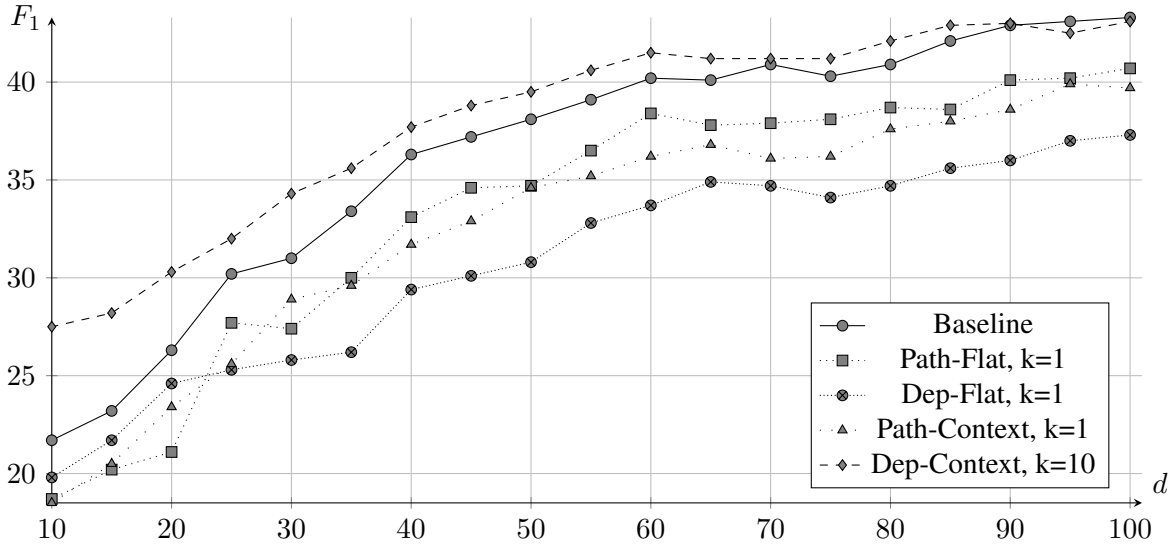


Figure 3: Learning curves (F_1 score for argument identification) with varying amounts of seed data d.

(2) a. *"This is largely a function of the much **smaller** [SD media]*$_{\text{entity}}$*."*

b. *"Plan for 8 **higher** quality [pics]*$_{\text{entity}}$ *or about 24 medium quality pics with internal memory ."*

Sentence 2a is the seed sentence, sentence 2b is the best sentence selected by the context-aware system. Though the two phrases *"smaller SD media"* and *"higher quality pics"* are a very good match, the word *"higher"* in sentence 2b does not express a product comparison. Instead, it describes a type of picture. Such uses are relatively frequent and often mistakenly chosen as expansion sentences. Such "false positives" mainly affect predicate identification, but errors in this first step are propagated through the pipeline.

Another type of error is caused by the non-aligned part of sentences. Sentences are sometimes rather long and contain other predicates besides the expanded predicate. Consider this example (3a seed, 3b context-aware system):

(3) a. *"That said, the **larger** LCD [screen]*$_{\text{aspect}}$ *is really an improvement."*

b. *"The **smaller** 2-inch [screen]*$_{\text{aspect}}$ *has higher resolution of 118,000 pixels!"*

The additional predicate *"higher"* in the expansion sentence is not detected, thereby creating a "false negative" example for the predicate identification classifier and the subsequent steps.

5 Conclusion

In this paper we investigate whether structural alignment, a semi-supervised method that has been successfully used for projecting SRL annotations to unlabeled sentences, can be adapted to the task of detecting comparisons. We find that some adjustments are necessary in order for the method to be applicable. First, we need to adapt the method of candidate selection to reflect that our arguments are further away from the predicate, while at the same time keeping the number of candidates manageable. Second, we need to adapt the similarity measure for scoring argument alignments to include context-aware measures. When we add the found expansion sentences to our training data, we can slightly improve over a non-expanded baseline in low-resource settings, i.e., when only a very small amount of training data in the desired domain or language is available.

There are many directions for future work. We have presented one possible context-aware similarity measure, but there are many other possibilities that can be explored. Two main issues are false positive and false negative predicates found by the expansion, the former being introduced by not detecting non-subjective usage of comparative words, the latter through other predicates besides the identified one being present in an expansion sentence. Doing subjectivity analysis to filter out non-comparative usages, and simplifying sentences or pre-selecting only short and simple sentences for expansion could improve results.

Acknowledgments

The work reported in this paper was supported by a Nuance Foundation grant.

References

Omri Abend, Roi Reichart, and Ari Rappoport. 2009. Unsupervised argument identification for semantic role labeling. In *Proceedings of ACL '09*, pages 28–36.

Anders Björkelund, Love Hafdell, and Pierre Nugues. 2009. Multilingual Semantic Role Labeling. In *Proceedings of CoNLL '09 Shared Task*, pages 43–48.

Hagen Fürstenau and Mirella Lapata. 2009. Semi-supervised semantic role labeling. In *Proceedings of EACL '09*, pages 220–228.

Hagen Fürstenau and Mirella Lapata. 2012. Semi-supervised semantic role labeling via structural alignment. *Computational Linguistics*, 38(1):135–171.

Murthy Ganapathibhotla and Bing Liu. 2008. Mining opinions in comparative sentences. In *Proceedings of COLING '08*, pages 241–248.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 238(3):245–288.

Feng Hou and Guo-hui Li. 2008. Mining Chinese comparative sentences by semantic role labeling. In *Proceedings of ICMLC '08*, pages 2563–2568.

Nitin Jindal and Bing Liu. 2006a. Identifying comparative sentences in text documents. In *Proceedings of SIGIR '06*, pages 244–251.

Nitin Jindal and Bing Liu. 2006b. Mining comparative sentences and relations. In *Proceedings of AAAI '06*, pages 1331–1336.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of WSDM '08*, pages 219–230, New York, NY, USA. ACM.

Richard Johansson and Pierre Nugues. 2007. Syntactic representations considered for frame-semantic analysis. In *Proceedings of TLT Workshop '07*, page 12.

Wiltrud Kessler and Jonas Kuhn. 2013. Detection of product comparisons - How far does an out-of-the-box semantic role labeling system take you? In *Proceedings of EMNLP '13*, pages 1892–1897.

Wiltrud Kessler and Jonas Kuhn. 2014. A corpus of comparisons in product reviews. In *Proceedings of LREC '14*.

Robert Swier and Suzanne Stevenson. 2005. Exploiting a verb lexicon in automatic semantic role labelling. In *Proceedings of HLT/EMNLP '05*, pages 883–890.

Kaiquan Xu, Stephen Shaoyi Liao, Jiexun Li, and Yuxia Song. 2011. Mining comparative opinions from customer reviews for competitive intelligence. *Decis. Support Syst.*, 50(4):743–754, March.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of EMNLP '04*, pages 88–94.