How Important is Syntactic Parsing Accuracy? An Empirical Evaluation on Rule-Based Sentiment Analysis

Carlos Gómez-Rodríguez · Iago Alonso-Alonso · David Vilares

Received: date / Accepted: date

Abstract Syntactic parsing, the process of obtaining the internal structure of sentences in natural languages, is a crucial task for artificial intelligence applications that need to extract meaning from natural language text or speech. Sentiment analysis is one example of application for which parsing has recently proven useful.

In recent years, there have been significant advances in the accuracy of parsing algorithms. In this article, we perform an empirical, task-oriented evaluation to determine how parsing accuracy influences the performance of a state-of-the-art rule-based sentiment analysis system that determines the polarity of sentences from their parse trees. In particular, we evaluate the system using four well-known dependency parsers, including both current models with state-of-the-art accuracy and more innacurate models which, however, require less computational resources.

The experiments show that all of the parsers produce similarly good results in the sentiment analysis task, without their accuracy having any relevant influence on the results. Since parsing is currently a task with a relatively high computational cost that varies strongly between algorithms, this suggests that sentiment analysis researchers and users should prioritize speed over accuracy when choosing a parser;

Carlos Gómez-Rodríguez has received funding from the European Research Council (ERC), under the European Union's Horizon 2020 research and innovation programme (FASTPARSE, grant agreement No 714150), Ministerio de Economía y Competitividad (FFI2014-51978-C2-2-R), and the Oportunius Program (Xunta de Galicia). Iago Alonso-Alonso was funded by an Oportunius Program Grant (Xunta de Galicia). David Vilares has received funding from the Ministerio de Educación, Cultura y Deporte (FPU13/01180) and Ministerio de Economía y Competitividad (FFI2014-51978-C2-2-R).

Carlos Gómez-Rodríguez · Iago Alonso-Alonso · David Vilares
FASTPARSE Lab, Grupo LyS, Departamento de Computación, Universidade da Coruña
Campus de A Coruña s/n, 15071, A Coruña, Spain
Tel.: +34 881 01 1396
Fax: +34 981 167 160
E-mail: carlos.gomez@udc.es, iago.alonso@udc.es, david.vilares@udc.es

and parsing researchers should investigate models that improve speed further, even at some cost to accuracy.

Keywords Syntactic Parsing · Sentiment Analysis · Natural Language Processing · Artificial Intelligence

1 Introduction

Having computers successfully understand the meaning of sentences in human languages is a long-standing key goal in artificial intelligence (AI). While full understanding is still far away, recent advances in the field of natural language processing (NLP) have made it possible to implement systems that can successfully extract relevant information from natural language text or speech. Syntactic parsing, the task of finding the internal structure of a sentence, is a key step in that process, as the predicate-argument structure of sentences encodes crucial information to understand their semantics. For example, a text mining system that needs to generate a report on customers' opinions about phones may find statements like "the iPhone is much better than the HTC 10" and "the HTC 10 is much better than the iPhone", which are identical in terms of the individual words that they contain. It is the syntactic structure – in this case, the subject and the attribute of the verb to be – that tells us which of the phones is preferred by the customer.

In recent years, parsing has gone from a merely promising basic research field to see widespread use in useful AI applications such as machine translation (Miceli Barone and Attardi 2015; Xiao et al 2016), information extraction (Song et al 2015; Yu et al 2015), textual entailment recognition (Padó et al 2015), learning for game AI agents (Branavan et al 2012) or sentiment analysis (Joshi and Penstein-Rosé 2009; Vilares et al 2015b,a). Meanwhile, researchers have produced improvements in parsing algorithms and models that have increased their accuracy, up to a point where some parsers have achieved levels comparable to agreement between experts on English newswire text (Berzak et al 2016), although this does not generalize to languages that present extra challenges for parsing (Farghaly and Shaalan 2009) or to noisy text such as tweets (Kong et al 2014). However, parsers consume significant computational resources, which can be an important concern in large-scale applications (Clark et al 2009), and the most accurate models often come at a higher computational cost (Andor et al 2016; Gómez-Rodríguez 2016). Therefore, an interesting question is how much influence parsing accuracy has on the performance of downstream applications, as this can be essential to make an informed choice of a parser to integrate in a given system.

In this article, we analyze this issue for sentiment analysis (SA), i.e., the use of natural language processing to extract and identify subjective information (opinions about relevant entities) from natural language texts. Sentiment analysis is one of the most relevant practical applications of NLP, it has been recently shown to benefit from parsing (Socher et al 2013; Vilares et al 2015b) and it is especially useful at a large scale (as millions of texts of potential interest for opinion extraction are generated every day in social networks), making the potential accuracy vs. speed tradeoff especially relevant.

For this purpose, we take a state-of-the-art syntax-based sentiment analysis system (Vilares et al 2017), which calculates the polarity of a text (i.e., whether it

expresses a positive, negative or neutral stance) relying on its dependency parse tree; and we test it with a set of well-known syntactic parsers, including models with state-of-the-art accuracy and others that are less accurate, but have a smaller computational cost, evaluating how the choice of parser affects the accuracy of the polarity classification. Our results show that state-of-the-art parsing accuracy does not provide additional benefit for this sentiment analysis task, as all of the parsers tested produce similarly good polarity classification accuracy (no statistically significant differences, all p-values ≥ 0.49). Therefore, our results suggest that it makes sense to use the fastest parsers for this task, even if they are not the most accurate.

The remainder of this article is organized as follows: we review the state of the art in syntactic parsing and syntax-based sentiment analysis in Section 2, we describe our experimental setup in Section 3, we report the results in Section 4, and discuss their implications in Section 5. Finally, Section 6 draws our conclusion and discusses possible avenues for future work.

2 Background

We now provide an overview of research in parsing and sentiment analysis that is relevant to this study.

2.1 Parsing

Different linguistic theories define different ways in which the syntactic structure of a sentence can be described. In particular, the overwhelming majority of natural language parsers in the literature adhere to one of two dominant representations. In *constituency grammar* (or *phrase structure grammar*), sentences are analyzed by breaking them up into segments called constituents, which are in turn decomposed into smaller constituents, as in the example of Figure 1. In *dependency grammar*, the syntax of a sentence is represented by directed binary relations between its words, called dependencies, which are most useful when labeled with their syntactic roles, such as subject and object, as in Figure 2. Each of these representation types provides different information about the sentence, and it is not possible to fully map constituency to dependency representations or vice versa (Kahane and Mazziotta 2015).

In this paper we will focus on dependency parsing, as it is the predominant representation used by most of the downstream AI applications mentioned above – with machine translation as an arguable exception, where constituent parsing is often used due to the adequacy of phrase structure grammar for modeling reordering of words between languages (DeNeefe and Knight 2009; Xiao et al 2016) –, and it is also the alternative used by the syntax-based SA system we will use in our experiments.

Most dependency parsing systems in the literature can be grouped into two broad categories (McDonald and Nivre 2007): *graph-based* and *transition-based* (shift-reduce) parsers.

Graph-based parsers use models that score dependency relations or groups of them, and perform a global search for a parse that will maximize the combined

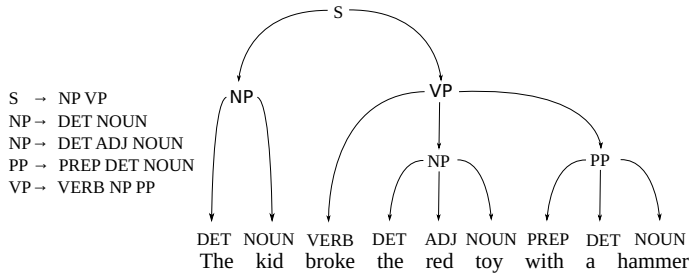


Fig. 1 A valid constituency parse for the sentence *'The kid broke the red toy with a hammer'*. The sentence is divided into constituents according to the constituency grammar defined at the left part of the picture

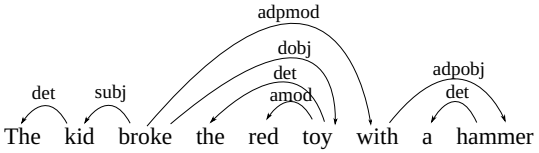


Fig. 2 A valid dependency parse for the sentence: *'The kid broke the red toy with a hammer'*. The sentence is represented as a graph of binary relations between words that represent the existing syntactic relation between them (e.g. *'kid'* is the subject of the verb *'broke'*)

score of all dependencies. Under the assumption of projectivity (i.e., that there are no crossing dependencies), there are several dynamic programming algorithms that perform exact search in cubic time (Eisner 1996; Gómez-Rodríguez et al 2008), but this restriction is not realistic in practice (Gómez-Rodríguez 2016). Unfortunately, exact inference has been shown to be intractable for models that support arbitrary non-projectivity, except under strong independence assumptions (McDonald and Satta 2007) which enable parsing in quadratic time with maximum spanning tree algorithms (McDonald et al 2005), but severely limit the expressivity of the feature models that can be used. This restriction can be avoided by using so-called *mildly non-projective* parsing algorithms, which support the overwhelming majority of non-projective analyses that can be found in real linguistic structures (Gómez-Rodríguez et al 2011; Cohen et al 2011; Pitler et al 2013); but they have super-cubic complexities that make them too slow for practical use. Another option is to forgo exact inference, using approximate inference algorithms with rich feature models instead. This is the approach taken by TurboParser (Martins et al 2010, 2013), which currently is the most popular graph-based parser as it can provide state-of-the-art accuracy with a reasonable computational cost.

Transition-based parsers are based on a state machine that builds syntactic analyses step by step, typically from left to right. A statistical or machine learning model scores each of the possible transitions to take at each state, and a search strategy is used to find a high-scoring sequence of transitions. The earlier approaches to transition-based parsing, like the MaltParser system (Nivre et al 2007) used greedy deterministic search for this purpose, which is especially fast, but is prone to obtain suboptimal solutions due to bad decisions at early stages that result in error propagation. This problem is alleviated by instead performing beam search (Zhang and Nivre 2011) or dynamic programming (Huang and Sagae

2010; Kuhlmann et al 2011) to explore transition sequences, but this increases the computational cost. Other alternatives that provide a good speed-accuracy tradeoff are selectional branching, which uses confidence estimates to decide when to employ a beam (Choi and McCallum 2013), or dynamic oracles, which reduce the error propagation in greedy search by exploring non-optimal transition sequences during training (Goldberg and Nivre 2012). In the last two years, several transition-based parsers have appeared that use neural networks as their scoring model (Chen and Manning 2014; Dyer et al 2015; Andor et al 2016), providing very good accuracy.

2.2 Parsing Evaluation

The standard metrics to evaluate the accuracy of a dependency parser are the unlabeled attachment score (UAS: the proportion of words that are attached to the correct head word by means of a dependency, regardless of its label), labeled attachment score (LAS: the proportion of words that are attached to the correct head by means of a dependency that has the correct label) and label accuracy (LA: the proportion of words that are assigned the correct dependency type). However, the performance of a parser in terms of such scores is not necessarily proportional to its usefulness for a given task, as not all dependencies in a syntactic analysis are equally useful in practice, or equally difficult to analyze (Nivre et al 2010; Bender et al 2011). Therefore, LAS, UAS and LA are of limited use for researchers and practitioners that work with downstream applications in NLP. For this purpose, it is more useful to perform task-oriented evaluation, i.e., to experiment with the parsers in the actual tasks for which they are going to be used (Volokh and Neumann 2012).

Such evaluations have been performed for some specific NLP tasks, namely information extraction (Miyao et al 2008; Buyko and Hahn 2010), textual entailment recognition (Yuret et al 2010; Volokh and Neumann 2012) and machine translation (Quirk and Corston-Oliver 2006; Goto et al 2011; Popel et al 2011). However, these comparisons are currently somewhat dated, as they were performed before the advent of the major advances in parsing accuracy of the current decade reviewed in Section 2.1, such as beam-search transition-based parsing, dynamic oracles, approximate variational inference (TurboParser) or neural network parsing. Even more importantly, these analyses provide very different results depending on each specific task and, to our knowledge, no evaluation of parsers has been performed for sentiment analysis, a task where a good speed-accuracy tradeoff is especially important due to its extensive applications to the Web and social networks.

2.3 Syntax-based Sentiment Analysis

A number of state-of-the-art models for SA using different morphological (Khan et al 2016a,b) and syntactic approaches have proven useful in recent years. Liu et al (2016) pointed out the benefits of syntactical approaches with respect to statistical models on opinion target extraction, such as domain independence, and propose two approaches to select a set of rules, that even being suboptimal, achieve better results than a state-of-the-art conditional random field supervised method.

Wu et al (2009) defined an approach to extract product features through phrase dependency parsing: they first combine the output of a shallow and a word-level dependency parser to then extract features and feed a support vector machine (SVM) with a novel tree kernel function. Their experimental results outperformed a number of bag-of-words baselines. Jia et al (2009) and Asmi and Ishaya (2012) defined a set of syntax-based rules for identifying and handling negation on natural language texts represented as dependency trees. They also pointed out the advantage of using this kind of methods with respect to traditional lexicon-based perspectives in tasks such as opinion mining or information retrieval. Poria et al (2014) posed a set of syntax-based patterns for a concept-level approach to determine how the sentiment flows from concept to concept, assuming that such concepts present in texts are represented as nodes of a dependency tree.

Joshi and Penstein-Rosé (2009) introduced the concept of generalized triplets, using them as features for a supervised classifier and showing its usefulness for subjectivity detection. Given a dependency triplet, the authors proposed to generalize the head or the dependent term (or even both at the same time) to its corresponding part-of-speech tag. Thus, the triplet *(car, modified, good)* could be generalized as *(NOUN, modifier, good)*, which can be useful to correctly classify similar triplets that did not appear in the training set (e.g. *(bicycle, modifier, good)* or *(job, modifier, good)*). In a similar line, Vilares et al (2015c) enriched the concept of generalized dependency triplets and showed that they can be exploited as features to feed a supervised SA system for polarity classification, as long as enough labeled data is available. The same authors (Vilares et al 2015b) proposed an unsupervised syntax-based approach for polarity classification on Spanish reviews represented as Ancora trees (Taulé et al 2008). They showed that their system outperforms the equivalent lexical-based approach (Taboada et al 2011). In this line, however, Taboada et al (2011) pointed out that one of the challenges when using parsing techniques for sentiment analysis is the need of fast parsers that are able to process in real-time the huge amount of information shared by users in social media.

With the recent success of deep learning, Socher et al (2013) syntactically annotated a sentiment treebank to then train a recursive neural network that learns how to apply semantic composition for relevant phenomena in SA, such as negation or *'but'* adversative clauses, over dependency trees. Kalchbrenner et al (2014) introduced a convolutional neural network for modeling sentences and used it for polarity classification among other tasks. Their approach does not explicitly rely on any parser, but the authors argue that one of the strengths of their model comes from the capability of the network to implicitly learn internal syntactic representations.

3 Materials and Methods

We now describe the systems, corpora and methods used for our task-oriented evaluation.

3.1 Parsing systems

- MaltParser: Introduced by Nivre et al (2007), this system can be used to train transition-based parsers with greedy deterministic search. Although its accuracy has fallen behind the state of the art, it is still widely used, probably owing to its maturity and solid documentation. Additionally, due to its greedy nature, MaltParser is very fast. Following common practice, we use it together with the feature optimization tool MaltOptimizer[1] (Ballesteros and Nivre 2012) to optimize the parameters and train a suitable model. The trained MaltParser model uses a standard arc-eager (transition-based) parsing algorithm, where at each step the movement to apply is selected among the set of possible transitions, previously scored by a linear model, which is faster than using models based on SVMs.
- TurboParser (Martins et al 2013): A graph-based parser that uses approximate variational inference with non-local features. It has become the most widely used graph-based parser, as it provides better speed and accuracy than previous alternatives. We use its default configuration, training a second-order non-projective parser with features for arcs, consecutive siblings and grandparents, using the AD3 algorithm as a decoder.
- YaraParser (Rasooli and Tetreault 2015): A recent transition-based parser, which uses beam search (Zhang and Nivre 2011) and dynamic oracles (Goldberg and Nivre 2012) to provide state-of-the-art accuracy. Its default configuration is used.
- Stanford RNN Parser (Chen and Manning 2014): The most popular among the recent wave of transition-based parsers that employ neural networks, it can achieve robust accuracy in spite of using greedy deterministic search. We use pretrained GloVe (Pennington et al 2014) embeddings as input to the parser: in particular, 50-dimensional word embeddings[2] trained on Wikipedia and the English Gigaword (Napoles et al 2012).

3.2 Parsing corpus

To train and evaluate the parsing accuracy of such parsers, we are using the English Universal Treebank v2.0 created by McDonald et al (2013). It is a mapping from the (constituency) Penn treebank (Marcus et al 1993) to a universal dependency grammar annotation. The choice of the treebank is due to the already existing predefined compositional operations in the SA system used for evaluation (see §3.3), that are intended for this type of universal guidelines. The corpus contains 39 833, 1 701 and 2 416 dependency trees for the training, development and test sets, respectively, and it represents one of the largest available treebanks for English.

[1] MaltParser often requires feature optimization to obtain acceptable results for the target language.

[2] http://nlp.stanford.edu/data/glove.6B.zip

3.3 Sentiment analysis system

For the task-oriented evaluation, we will rely on UUUSA, the universal, unsupervised, uncovered approach for sentiment analysis described by (Vilares et al 2017), which is based on syntax and the concept of *compositional operations*. Briefly, given a text represented as a dependency tree, a *compositional operation* defines how a node of the tree modifies the semantic orientation (a real value representing a polarity and its strength) of a different branch or node, based on features such as its word form, part-of-speech tag or dependency type, without any limitation in terms of its location inside such tree. The associated system queues operations and propagates them through the tree, until the moment they must be dequeued and applied to their target. The model has outperformed other state-of-the-art lexicon-based methods on a number of corpora and languages, showing the advantages of using syntactic information for sentiment analysis. Due to the way the system works, in such a way that the application of the operation relies on previously assigning dependency types and heads correctly, it also constitutes a proper environment to test how parsing accuracy affects polarity classification.

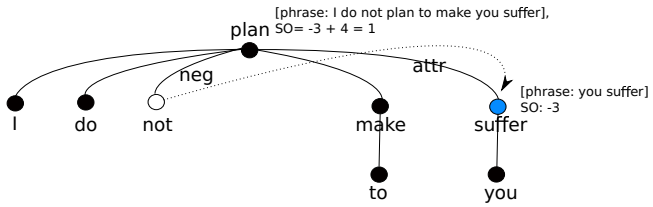
The system already includes a predefined set of universal syntactic operations, that we are using in this study to determine the importance of parsing accuracy. For the sake of brevity, we are not detailing how the system computes the semantic orientation of the trees, but we specify which universal dependencies UUUSA is relying on to identify relevant linguistic phenomena that should trigger a compositional operation. To apply an operation, usually a dependency type must match at the node a branch is rooted at. The existing set of predefined operations that we are considering involve phenomena such as:

- *Intensification*: A branch amplifies or decreases the semantic orientation of its head node or other branch (that must be labeled with the *acomp* (adjectival complement) dependency type). The intensifier branch must be labeled as one of these three dependency types: *advmod* (adverb modifier), *amod* (adjective modifier), *nmod* (noun modifier). Dependencies are relevant in this case because they help avoid false positive cases when applying intensification (e.g. in *'It is huge'*, *'huge'* should be (probably) a positive adjective, meanwhile in *'I have huge problems'* it acts as an intensifier as it is an adjective modifier of the negative word *'problems'*, and in *'I have huge exciting news'* it acts again as an intensifier, but of a positive term).
- *'But' clauses*: To trigger this compositional operation, which decreases the relevance of the semantic orientation of the main sentence, the dependent branch rooted at *'but'* must be labeled as *cc*.
- *Negation*: The negating terms, that might shift the sentiment of other branches, are labeled in a dependency tree with the dependency type *neg*.
- *'If'*: We also include experiments using the proposed rule in Vilares et al (2017) for the *'if'* clause, which is labeled with the *mark* dependency type, assuming that the part of the sentence under the scope of influence of the conditional clause should be ignored.

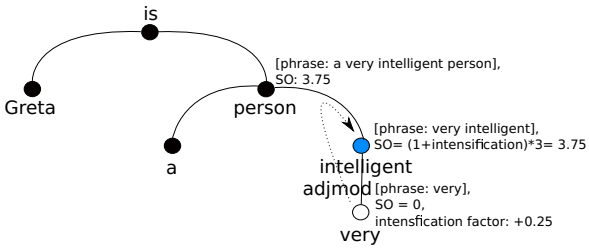
Therefore, the accuracy obtained by UUUSA on the sentiment corpora is related to the parsing accuracy: a LAS of zero makes it impossible to trigger any compositional operation, since no dependency type would match; obtaining as

output a global polarity which is the result of simply summing the semantic orientation of individual words.

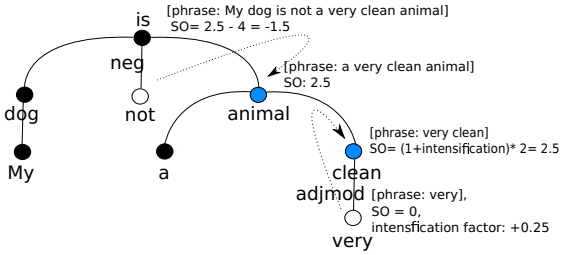
Figure 3.a) and Figure 3.b) illustrate two simple examples where part-of-speech tags, dependencies and types play a relevant role to accurately capture the semantic orientation of the sentence. Additionally, Figure 3.c) illustrates with an additional example how semantic composition is managed when a negation and an intensification appear in the same sentence and affect the same subjective word.



a) '*I do not plan to make you suffer*'



b) '*Greta is a very intelligent person*'



b) '*My dog is not a very clean animal*'

Fig. 3 Analysis of three sentences using the UUUSA approach. The analysis corresponds to a post-order recursive traversal. Semantic orientation, intensification and negation values are orientative. At each level, we show for the relevant nodes, those playing any role in the computation of the semantic orientation, the corresponding phrase rooted at that node, and its corresponding semantic orientation (SO), once compositional operations have been applied at that level. Sentence a) shows an example where the semantic scope of the negation is non-local, but thanks to dependency parsing and syntactic rules, the system can accurately identify such scope and shift the semantic orientation coming from that branch. Note that, if either the dependency type *neg* or *attr* were assigned incorrectly, the calculation of the SO would be wrong. Sentence b) illustrates how the term *'very'* increases the semantic orientation of its head. It is important to remark that if the dependency type *adjmod* were assigned incorrectly, the analysis would be again unaccurate. Sentence c) illustrates a more complex compositional example, where first, the intensifier *'very'* amplifies the semantic orientation of the word *'clean'*, and the negating word *'not'* shifts the sentiment rooted at the phrase *'a very clean animal'*

We chose this system among others for three main reasons:

1. It supports separate compositional operations to address very specific linguistic phenomena, which can be enabled or disabled individually. This gives us great flexibility to carry out experiments including and excluding a number of linguistic constructions, allowing us to determine how relevant parsing accuracy is to tackle each of them.
2. It is a modular system where the parser is an independent component that can be swapped with another parser, allowing us to use it for task-oriented evaluation of various parsers. This contrasts with Socher et al (2013), a system that also uses syntax, but where the parsing process is tightly woven with the sentiment analysis process (a neural network architecture is trained to perform both tasks at the same time) so that it is not possible to use it with the output of external parsers.
3. Symbolic or knowledge-based systems like this perform robustly across different datasets and domains, which we cannot guarantee for the case of many machine learning models, that do not generalize so well (Aue and Gamon 2005; Taboada et al 2011; Vilares et al 2017).

3.4 Sentiment analysis corpora

Three standard corpora for document- and sentence-level sentiment analysis are used for the extrinsic evaluation:

- Taboada and Grieve (2004) corpus: A general-domain dataset composed of 400 long reviews (50% positive, 50% negative) about different topics (e.g. washing-machines, books or computers).
- Pang and Lee (2004) corpus: A collection of 2 000 long movie reviews (50% positive, 50% negative).
- Pang and Lee (2005) corpus: A collection of short (i.e. single-sentence) movie reviews. We relied on the test split used by Socher et al (2013), removing the neutral ones, as they did, for the binary classification task (1 821 subjective sentences: $\sim$49% positive, $\sim$51% negative).

3.5 Experimental methodology

The aim of our experiments is to show how parsing accuracy influences polarity classification, following a task-oriented evaluation. To do so, we first compare the performance of different parsers on a standard treebank test set and metrics. We then extrinsically evaluate the performance of such parsers by parsing sentiment corpora, and using the obtained parse trees to determine the polarity of the texts in the corpora by means of a state-of-the-art syntax-based model. The performance of this model relies on previous correct assignment of dependency types and heads, to be able to handle relevant linguistic phenomena for the purpose at hand (e.g. intensification, *'but'* clauses or negation). This makes it possible to relate parsing and syntax-based sentiment performance.

3.6 Hardware and software used in the experiments

Experiments were carried in a Dell XPS 8500 Intel Core i7 @ 3.4GHz and 16GB of RAM. Operating system was Ubuntu 14.04 64 bits.

4 Results

Table 1 shows the performance obtained by the different parsers according to the standard metrics: LAS, UAS and LA. Table 2 illustrates how much time each parser consumes to analyze the Pang and Lee (2005) corpus, and the total time once the SA system is run on it.

Tables 3, 4 and 5 show the accuracy obtained by UUUSA on different sentiment corpora, when the output of each of the parsers is used as input to the syntax-based sentiment analysis system.[3] We take accuracy as the reference metric for the SA systems, because it is the most suitable metric in this case, since the three corpora are balanced. In particular, we compare the performance when no syntactic rules are used (which would be equivalent to a lexicon-based system that only sums the semantic orientation of individual words), with respect to the one obtained when different rules are added. The aim is to determine if different parsers manage relevant linguistic phenomena in a different way.

Finally, Figure 4 relates the LAS performance on the test set of the universal treebank with respect to the accuracy obtained by UUUSA, when we artificially reduce the training set size to simulate a low-accuracy parsing setting, as could happen in low-resource languages.

[3] The results obtained in these corpora are slightly different from the ones reported by Vilares et al (2017), due to the different tokenization techniques used in this work.

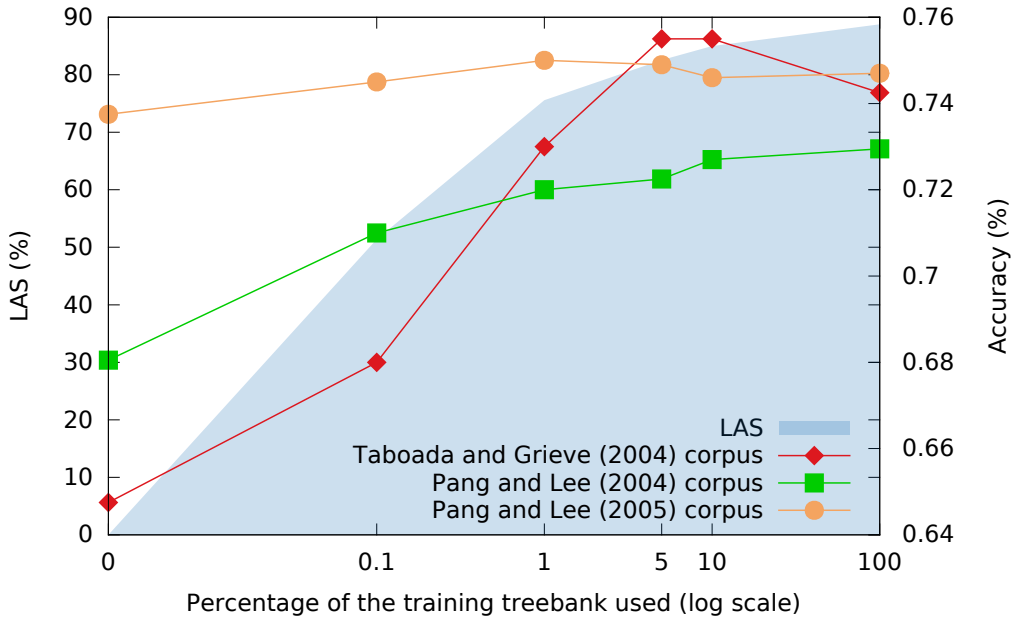


Fig. 4 Relationship between LAS (area graphic, left y-axis) and accuracy in different sentiment corpora (line graphics, right y-axis), using the Stanford RNN parser (Chen and Manning 2014) trained with different portions (%) of the training treebank (x-axis). The plot shows that using a larger training treebank improves LAS, but does not necessarily increase the UUUSA sentiment accuracy, especially when more than a 5 or 10% of such treebank is used to train such parser.

5 Discussion

The results illustrated in Tables 1 and 2 indicate the relationship between the parsing time and accuracy. The slower parsers (Martins et al 2013; Rasooli and Tetreault 2015) tend to obtain a better performance, meanwhile the faster ones (Nivre et al 2007; Chen and Manning 2014) attain worse LAS, UAS and LA. This fact is expected, as there is a well-known tradeoff between speed and accuracy in the spectrum of parsing algorithms, with one extreme at greedy search approaches that scan and parse the sentence in a single pass but are prone to error propagation, and the other at exact search algorithms that guarantee finding the highest-scoring parse under a rich statistical model, but are prohibitively slow (Choi and McCallum 2013; Volokh 2013; Gómez-Rodríguez 2016). The tendency remains when looking at the performance on individual dependency types (where also the head is assigned correctly).

However, a better LAS or UAS does not necessarily translate into a higher sentiment accuracy, which is shown in Tables 3, 4 and 5. In most cases, the performance obtained by the different parsers under the same sets of rules is practically equivalent. To confirm this statistically, we applied chi-squared significance tests to compare the outputs obtained using the different parsers for each given dataset and set of sentiment rules. No significant differences in sentiment accuracy were found in any of these experiments, which reinforces our conclusion. The minimum

Table 1 Performance (LAS, UAS and LA) of the parsers on the English Universal treebank test set. We also detail the performance in terms of Precision (P) and Recall (R) for the dependency types that are playing a role in the predefined compositional operations of UUUSA. The subscripts indicate the rank of the parser with respect to the others, given a particular metric

Metric	MaltParser	Stanford RNN parser	TurboParser	YaraParser
LAS	88.35_4	88.77_3	91.36_2	$\mathbf{91.84}_1$
UAS	90.27_4	90.47_3	93.29_2	$\mathbf{93.34}_1$
LA	93.01_4	93.59_3	95.02_2	$\mathbf{95.72}_1$
P(*acomp*)	88.66_3	88.31_4	88.75_2	$\mathbf{91.03}_1$
R(*acomp*)	90.29_3	89.24_4	91.08_2	$\mathbf{93.18}_1$
P(*advmod*)	83.18_3	82.38_4	84.96_2	$\mathbf{85.85}_1$
R(*advmod*)	84.04_3	81.97_4	$\mathbf{85.46}_1$	85.22_2
P(*amod*)	95.45_3	95.01_4	$\mathbf{96.25}_1$	96.23_2
R(*amod*)	95.45_3	95.40_4	96.30_2	$\mathbf{96.60}_1$
P(*attr*)	87.17_3	86.00_4	89.00_2	$\mathbf{94.88}_1$
R(*attr*)	88.63_3	86.29_4	91.97_2	$\mathbf{92.98}_1$
P(*cc*)	77.06_4	77.68_3	$\mathbf{83.56}_1$	83.46_2
R(*cc*)	76.95_4	77.02_3	83.82_2	$\mathbf{83.97}_1$
P(*mark*)	83.44_4	84.21_3	85.15_2	$\mathbf{88.47}_1$
R(*mark*)	83.44_4	88.31_3	90.26_2	$\mathbf{92.21}_1$
P(*neg*)	92.99_2	92.62_4	$\mathbf{94.77}_1$	92.68_3
R(*neg*)	94.72_2	93.48_4	$\mathbf{95.65}_1$	94.41_3
P(*nmod*)	80.66_4	82.17_3	84.45_2	$\mathbf{85.38}_1$
R(*nmod*)	79.67_2	78.66_4	79.47_3	$\mathbf{80.69}_1$

Table 2 Average, maximum and minimum execution time (seconds) out of 5 runs on the Pang and Lee (2005) test set. We also include the total execution time, after the SA system has been run on the Pang and Lee (2005) corpus

Parser	Average	Minimum	Maximum	Average + UUUSA time ($\sim$ **1.2 sec**)
MaltParser	**4.02**	**3.83**	**4.23**	**5.22**
Stanford RNN Parser	5.99	5.82	6.23	7.19
Turbo Parser	97.34	94.79	99.23	98.54
Yara Parser	39.85	38.94	41.23	41.05

Table 3 Accuracy on the Pang and Lee (2004) corpus considering different subsets of rules

Parser	All	None	Intensification	*'but'*	*'if'*	Negation
MaltParser	72.75	68.05	70.35	67.75	68.20	69.80
Stanford RNN Parser	**72.95**	68.05	**70.60**	67.85	68.35	**70.30**
Turbo Parser	72.20	68.05	70.35	**67.95**	68.25	69.60
Yara Parser	72.00	68.05	70.30	67.85	**68.55**	70.15

Table 4 Accuracy on the Pang and Lee (2005) corpus considering different subsets of rules

Parser	All	None	Intensification	*'but'*	*'if'*	Negation
MaltParser	**74.79**	73.75	**74.41**	73.86	73.75	74.14
Stanford RNN Parser	74.68	73.75	74.35	73.86	**73.97**	**74.19**
Turbo Parser	74.57	73.75	**74.41**	**73.92**	73.86	**74.19**
Yara Parser	74.68	73.75	**74.41**	73.86	**73.97**	73.92

Table 5 Accuracy on the Taboada and Grieve (2004) corpus considering different subsets of rules

Parser	All	None	Intensification	*'but'*	*'if'*	Negation
MaltParser	74.00	64.75	66.25	**64.75**	64.00	72.25
Stanford RNN Parser	74.25	64.75	66.25	**64.75**	**64.25**	**72.75**
Turbo Parser	**75.00**	64.75	**66.50**	64.50	63.75	**72.75**
Yara Parser	73.50	64.75	**66.50**	64.50	63.75	72.00

p-value obtained was 0.49. It is important to remark that this is very different from stating that parsing is not relevant for SA. In the case of UUUSA, Vilares et al (2017) already showed that their syntax-based SA approach is able to beat purely lexicon-based methods on a number of languages. In this line, Tables 3, 4 and 5 also show that the sets of syntactic rules outperform the baseline that does not use any syntactic-based rules ('None' column) in almost all cases, proving again that syntax-based rules are useful to handle relevant linguistic phenomena in the field of SA.

The specific reasons that explain why the choice of syntactic parsing algorithm does not significantly affect accuracy lie out of the scope of our empirical work, as they require an exhaustive linguistic analysis. In view of the data, possible factors that may contribute are the following:

- Low difficulty of some of the most decisive dependencies involved: as can be seen in Table 1, even the least accurate parsers analyzed are obtaining well over 92% precision and recall in adjectival modifiers (amod) and negations (neg), which are crucial for handling intensification and negation. This is likely because these tend to be short-distance dependencies, which are easier to parse (McDonald and Nivre 2007), and are common so they do not suffer from training sparsity problems. Thus, a highly accurate parser is not needed to detect these particular dependencies correctly.
- Redundancy in sentences: a sentence may include several expressions of sentiment, so that even if the parse tree contains inaccuracies in a part of the sentence, we may still be able to extract the correct sentiment from the rest. This can be especially frequent in long sentences, which are the most difficult to parse (McDonald and Nivre 2007).
- Irrelevance of fine-grained distinctions: in some cases, the parser provides more information than is strictly needed to evaluate the sentiment of a sentence. For example, the UUUSA rule for intensifiers works in the same way for adverbial modifiers (advmod), adjectival modifiers (amod) or nominal modifiers (nmod). Thus, if a parser mistakes e.g. an advmod for an amod, this counts as a parsing error, but has no influence in the sentiment output.

However, verifying and quantifying the influence of each of these factors remains as an open question, which we would like to explore in the near future.

An interesting conclusion that could be extracted from these results is that parsing should prioritize speed over accuracy for syntax-based polarity classification. We draw Figure 4 to reinforce this hypothesis. The figure illustrates how LAS and sentiment accuracy vary when training the Stanford RNN parser (Chen and Manning 2014) with different training data size. To do so, we trained a number of parsers using the first x% of the training treebank. As expected, it was observed that adding more training data increased the LAS obtained by the parser. How-

ever, this same tendency did not remain with respect to sentiment accuracy, which remains stable once LAS reaches an acceptable level. Based on empirical evaluation, sentiment accuracy stops increasing when using the first 5% (82.57% LAS) or 10% (84.99% LAS) of the English Universal training treebank, with which it is possible to already obtain a performance close to the state of the art (88.77% when using the whole training treebank). On the other hand, there is a clear increasing tendency when $x < 5$, because in those cases the LAS is still not good enough (using the first 0.1% and 1% of the training treebank we only are able to achieve a LAS of 51.39% and 75.58%, respectively).

6 Conclusions

In this article, we have carried out a task-oriented empirical evaluation to determine the relevance of parsing accuracy on the primary challenge of sentiment analysis: polarity classification. We chose English as the target language and trained a number of standard and freely available parsers on the Universal Dependency Treebank v2.0 (McDonald et al 2013). The output of such parsers on different standard sentiment corpora is then used as input for a state-of-the-art and syntax-based system that aims to classify the polarity of those texts. Experimental results let us draw two interesting and promising conclusions: (1) a better labeled/unlabeled attachment score on parsing does not necessarily imply a significantly better accuracy on polarity classification when using syntax-based algorithms and (2) parsing for sentiment analysis should focus on speed instead of accuracy, as a LAS of around 80% (which we obtained in the experiments by using only the first 10% of the training treebank) is already good enough to fully take advantage of dependency trees and exploit syntax-based rules. Using larger training portions produces increases in the labeled attachment score up to the maximum value of close to 92% that we obtained with the most accurate parser, but the performance for sentiment accuracy remains stable. Hence, there is no reason to use a slower parser to maximize LAS as long as one is above said "good enough" threshold for sentiment analysis, which is clearly surpassed by all the parsers tested.

Based on the results, we believe there is room for improvements. We plan to design algorithms for faster parsing (Volokh 2013), prioritizing speed over accuracy. We also would like to explore the influence of parsing accuracy on other high-level tasks analysis, such as aspect extraction (Wu et al 2009) or question answering (Rajpurkar et al 2016), where dependencies have played an important role.

References

Andor D, Alberti C, Weiss D, Severyn A, Presta A, Ganchev K, Petrov S, Collins M (2016) Globally normalized transition-based neural networks. arXiv 1603.06042 [cs.CL], URL `http://arxiv.org/abs/1603.06042`

Asmi A, Ishaya T (2012) Negation identification and calculation in sentiment analysis. In: The Second International Conference on Advances in Information Mining and Management, pp 1–7

Ballesteros M, Nivre J (2012) Maltoptimizer: A system for maltparser optimization. In: Chair) NCC, Choukri K, Declerck T, Dogan MU, Maegaard B, Mariani J, Moreno A, Odijk J, Piperidis S (eds) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA), Istanbul, Turkey

Bender EM, Flickinger D, Oepen S, Zhang Y (2011) Parser evaluation over local and non-local deep dependencies in a large corpus. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, Scotland, UK., pp 397–408, URL http://www.aclweb.org/anthology/D11-1037

Berzak Y, Huang Y, Barbu A, Korhonen A, Katz B (2016) Bias and agreement in syntactic annotations. arXiv 1605.04481 [cs.CL], URL https://arxiv.org/abs/1605.04481

Branavan SRK, Silver D, Barzilay R (2012) Learning to win by reading manuals in a monte-carlo framework. J Artif Int Res 43(1):661–704, URL http://dl.acm.org/citation.cfm?id=2387915.2387932

Buyko E, Hahn U (2010) Evaluating the impact of alternative dependency graph encodings on solving event extraction tasks. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Cambridge, MA, pp 982–992, URL http://www.aclweb.org/anthology/D10-1096

Chen D, Manning C (2014) A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp 740–750, URL http://www.aclweb.org/anthology/D14-1082

Choi JD, McCallum A (2013) Transition-based dependency parsing with selectional branching. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria, pp 1052–1062, URL http://www.aclweb.org/anthology/P13-1104

Clark S, Copestake A, Curran JR, Zhang Y, Herbelot A, Haggerty J, Ahn BG, Wyk CV, Roesner J, Kummerfeld J, Dawborn T (2009) Large-scale syntactic processing: Parsing the web. Tech. rep., Johns Hopkins University

Cohen SB, Gómez-Rodríguez C, Satta G (2011) Exact inference for generative probabilistic non-projective dependency parsing. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp 1234–1245, URL http://www.aclweb.org/anthology/D11-1114

DeNeefe S, Knight K (2009) Synchronous tree adjoining machine translation. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, pp 727–736, URL http://www.aclweb.org/anthology/D/D09/D09-1076

Dyer C, Ballesteros M, Ling W, Matthews A, Smith NA (2015) Transition-based dependency parsing with stack long short-term memory. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Beijing, China, pp 334–343, URL http://www.aclweb.org/anthology/P15-1033

Eisner J (1996) Three new probabilistic models for dependency parsing: An exploration. In: Proceedings of the 16th International Conference on Computational Linguistics (COLING-96), San Francisco, CA, USA, pp 340–345

Farghaly A, Shaalan K (2009) Arabic natural language processing: Challenges and solutions. ACM Transactions on Asian Language Information Processing (TALIP) 8(4):14:1–14:22, DOI 10.1145/1644879.1644881, URL http://doi.acm.org/10.1145/1644879.1644881

Goldberg Y, Nivre J (2012) A dynamic oracle for arc-eager dependency parsing. In: Proceedings of the 24th International Conference on Computational Linguistics (COLING), Association for Computational Linguistics, pp 959–976, URL http://aclweb.org/anthology/C/C12/C12-1059.pdf

Gómez-Rodríguez C (2016) Restricted non-projectivity: Coverage vs. efficiency. Comput Linguist 42(4):809–817, DOI 10.1162/COLI_a_00267, URL http://dx.doi.org/10.1162/COLI_a_00267

Gómez-Rodríguez C, Carroll J, Weir D (2008) A deductive approach to dependency parsing. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL'08:HLT), Association for Computational Linguistics, pp 968–976, URL http://www.aclweb.org/anthology/P/P08/P08-1110

Gómez-Rodríguez C, Carroll JA, Weir DJ (2011) Dependency parsing schemata and mildly non-projective dependency parsing. Computational Linguistics 37(3):541–586

Goto I, Utiyama M, Onishi T, Sumita E (2011) A comparison study of parsers for patent machine translation. In: Proceedings of the 13th Machine Translation Summit (MT Summit XIII), International Association for Machine Translation, pp 448–455, URL http://www.mt-archive.info/MTS-2011-Goto.pdf

Huang L, Sagae K (2010) Dynamic programming for linear-time incremental parsing. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10, pp 1077–1086, URL http://portal.acm.org/citation.cfm?id=1858681.1858791

Jia L, Yu C, Meng W (2009) The effect of negation on Sentiment Analysis and Retrieval Effectiveness. In: CIKM'09 Proceeding of the 18th ACM conference on Information and knowledge management, ACM, ACM Press, Hong Kong, pp 1827–1830

Joshi M, Penstein-Rosé C (2009) Generalizing dependency features for opinion mining. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, Association for Computational Linguistics, Stroudsburg, PA, USA, ACLShort '09, pp 313–316

Kahane S, Mazziotta N (2015) Syntactic polygraphs. a formalism extending both constituency and dependency. In: Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015), Association for Computational Linguistics, Chicago, USA, pp 152–164, URL http://www.aclweb.org/anthology/W15-2313

Kalchbrenner N, Grefenstette E, Blunsom P (2014) A Convolutional Neural Network for Modelling Sentences. In: The 52nd Annual Meeting of the Association for Computational Linguistics. Proceedings of the Conference. Volume 1: Long Papers, ACL, Baltimore, Maryland, USA, pp 655–665

Khan FH, Qamar U, Bashir S (2016a) esap: A decision support framework for enhanced sentiment analysis and polarity classification. Information Sciences 367:862–873

Khan FH, Qamar U, Bashir S (2016b) Swims: Semi-supervised subjective feature weighting and intelligent model selection for sentiment analysis. Knowledge-Based Systems 100:97–111

Kong L, Schneider N, Swayamdipta S, Bhatia A, Dyer C, Smith NA (2014) A dependency parser for tweets. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, pp 1001–1012, URL http://www.aclweb.org/anthology/D14-1108

Kuhlmann M, Gómez-Rodríguez C, Satta G (2011) Dynamic programming algorithms for transition-based dependency parsers. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL 2011), Association for Computational Linguistics, Portland, Oregon, USA, pp 673–682, URL http://www.aclweb.org/anthology/P11-1068

Liu Q, Gao Z, Liu B, Zhang Y (2016) Automated rule selection for opinion target extraction. Konwledge-Based Systems 104:74–88

Marcus MP, Marcinkiewicz MA, Santorini B (1993) Building a large annotated corpus of english: The penn treebank. Computational linguistics 19(2):313–330

Martins A, Smith N, Xing E, Aguiar P, Figueiredo M (2010) Turbo parsers: Dependency parsing by approximate variational inference. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Cambridge, MA, pp 34–44, URL http://www.aclweb.org/anthology/D10-1004

Martins A, Almeida M, Smith NA (2013) Turning on the turbo: Fast third-order non-projective turbo parsers. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Sofia, Bulgaria, pp 617–622, URL http://www.aclweb.org/anthology/P13-2109

McDonald R, Nivre J (2007) Characterizing the errors of data-driven dependency parsing models. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp 122–131

McDonald R, Satta G (2007) On the complexity of non-projective data-driven dependency parsing. In: IWPT 2007: Proceedings of the 10th International Conference on Parsing Technologies, pp 121–132

McDonald R, Pereira F, Ribarov K, Hajič J (2005) Non-projective dependency parsing using spanning tree algorithms. In: HLT/EMNLP 2005: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp 523–530

McDonald R, Nivre J, Quirmbach-brundage Y, Goldberg Y, Das D, Ganchev K, Hall K, Petrov S, Zhang H, Täckström O, Bedini C, Castelló N, Lee J (2013) Universal Dependency Annotation for Multilingual Parsing. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp 92–97

Miceli Barone AV, Attardi G (2015) Non-projective dependency-based pre-reordering with recurrent neural network for machine translation. In: Proceedings of the 53rd Annual

Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Beijing, China, pp 846–856, URL http://www.aclweb.org/anthology/P15-1082

Miyao Y, Sætre R, Sagae K, Matsuzaki T, Tsujii J (2008) Task-oriented evaluation of syntactic parsers and their representations. In: Proceedings of ACL-08: HLT, Association for Computational Linguistics, Columbus, Ohio, pp 46–54, URL http://www.aclweb.org/anthology/P/P08/P08-1006

Napoles C, Gormley M, Van Durme B (2012) Annotated gigaword. In: Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction, Association for Computational Linguistics, pp 95–100

Nivre J, Hall J, Nilsson J, Chanev A, Eryiğit G, Kübler S, Marinov S, Marsi E (2007) Maltparser: A language-independent system for data-driven dependency parsing. Natural Language Engineering 13:95–135

Nivre J, Rimell L, McDonald R, Gómez Rodríguez C (2010) Evaluation of dependency parsers on unbounded dependencies. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Association for Computational Linguistics, pp 833–841, URL http://www.aclweb.org/anthology/C10-1094

Padó S, Noh TG, Stern A, Wang R, Zanoli R (2015) Design and realization of a modular architecture for textual entailment. Natural Language Engineering 21(2):167–200

Pang B, Lee L (2004) A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp 271–278

Pang B, Lee L (2005) Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp 115–124

Pennington J, Socher R, Manning CD (2014) Glove: Global Vectors for Word Representation. In: EMNLP, vol 14, pp 1532–1543

Pitler E, Kannan S, Marcus M (2013) Finding optimal 1-endpoint-crossing trees. Transactions of the Association of Computational Linguistics 1:13–24, URL http://aclweb.org/anthology/Q13-1002

Popel M, Mareček D, Green N, Zabokrtsky Z (2011) Influence of parser choice on dependency-based mt. In: Proceedings of the Sixth Workshop on Statistical Machine Translation, Association for Computational Linguistics, Edinburgh, Scotland, pp 433–439, URL http://www.aclweb.org/anthology/W11-2153

Poria S, Cambria E, Winterstein G, Huang GB (2014) Sentic patterns: Dependency-based rules for concept-level sentiment analysis. Knowledge-Based Systems 69:45–63

Quirk C, Corston-Oliver S (2006) The impact of parse quality on syntactically-informed statistical machine translation. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Sydney, Australia, pp 62–69, URL http://www.aclweb.org/anthology/W06-1608

Rajpurkar P, Zhang J, Konstantin L, Liang P (2016) SQuAD: 100,000+ Questions for Machine Comprehension of Text. arXiv preprint arXiv:160605250

Rasooli MS, Tetreault JR (2015) Yara parser: A fast and accurate dependency parser. CoRR abs/1503.06733, URL http://arxiv.org/abs/1503.06733

Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng A, Potts C (2013) Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: EMNLP 2013. 2013 Conference on Empirical Methods in Natural Language Processing. Proceedings of the Conference, ACL, Seattle, Washington, USA, pp 1631–1642

Song M, Kim WC, Lee D, Heo GE, Kang KY (2015) PKDE4J: entity and relation extraction for public knowledge discovery. Journal of Biomedical Informatics 57:320–332, DOI 10.1016/j.jbi.2015.08.008, URL http://dx.doi.org/10.1016/j.jbi.2015.08.008

Taboada M, Grieve J (2004) Analyzing appraisal automatically. In: Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text (AAAI Technical Report SS0407), Stanford University, CA, AAAI Press, pp 158–161

Taboada M, Brooke J, Tofiloski M, Voll K, Stede M (2011) Lexicon-based methods for sentiment analysis. Computational Linguistics 37(2):267–307

Taulé M, Martí MA, Recasens M (2008) AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In: Calzolari N, Choukri K, Maegaard B, Mariani J, Odjik J, Piperidis S,

Tapias D (eds) Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, pp 96–101

Vilares D, Alonso MA, Gómez-Rodríguez C (2015a) A linguistic approach for determining the topics of Spanish Twitter messages. Journal of Information Science 41(02):127–145

Vilares D, Alonso MA, Gómez-Rodríguez C (2015b) A syntactic approach for opinion mining on Spanish reviews. Natural Language Engineering 21(01):139–163

Vilares D, Alonso MA, Gómez-Rodríguez C (2015c) On the usefulness of lexical and syntactic processing in polarity classification of Twitter messages. Journal of the Association for Information Science Science and Technology 66(9):1799–1816

Vilares D, Gómez-Rodríguez C, Alonso MA (2017) Universal, unsupervised (rule-based), uncovered sentiment analysis. Knowledge-Based Systems 118:45–55, DOI https://doi.org/10.1016/j.knosys.2016.11.014

Volokh A (2013) Performance-oriented dependency parsing. Doctoral dissertation, Saarland University, Saarbrücken, Germany

Volokh A, Neumann G (2012) Task-oriented dependency parsing evaluation methodology. In: IEEE 13th International Conference on Information Reuse & Integration, IRI 2012, Las Vegas, NV, USA, August 8-10, 2012, pp 132–137, DOI 10.1109/IRI.2012.6303001, URL http://dx.doi.org/10.1109/IRI.2012.6303001

Wu Y, Zhang Q, Huang X, Wu L (2009) Phrase Dependency Parsing for Opinion Mining. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, ACL, Singapore, pp 1533–1541

Xiao T, Zhu J, Zhang C, Liu T (2016) Syntactic skeleton-based translation. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA., pp 2856–2862, URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11933

Yu M, Gormley MR, Dredze M (2015) Combining word embeddings and feature embeddings for fine-grained relation extraction. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, pp 1374–1379, URL http://www.aclweb.org/anthology/N15-1155

Yuret D, Han A, Turgut Z (2010) Semeval-2010 task 12: Parser evaluation using textual entailments. In: Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Uppsala, Sweden, pp 51–56, URL http://www.aclweb.org/anthology/S10-1009

Zhang Y, Nivre J (2011) Transition-based dependency parsing with rich non-local features. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2, pp 188–193, URL http://dl.acm.org/citation.cfm?id=2002736.2002777