Grammatical Feature Engineering for fine-grained IR tasks

Danilo Croce and Roberto Basili

Department of Enterprise Engineering
University of Roma, Tor Vergata
{croce,basili}@info.uniroma2.it

Abstract. Information Retrieval tasks include nowadays more and more complex information in order to face contemporary challenges such as Opinion Mining (OM) or Question Answering (QA). These are examples of tasks where complex linguistic information is required for reasonable performances on realistic data sets. As natural language learning is usually applied to these tasks, rich structures, such as parse trees, are critical as they require complex resources and accurate pre-processing. In this paper, we show how good quality language learning methods can be applied to the above tasks by using grammatical representations simpler than parse trees. These features are here shown to achieve the state-of-art accuracy in different IR tasks, such as OM and QA.

1 Syntactic modeling of linguistic features in Semantic Tasks

Information Retrieval faces nowadays contemporary challenges such as Sentiment Analysis (SA) or Question Answering (QA), that are tight to complex and fine grained linguistic information. The traditional view in IR that represents the meaning of documents just according to the words that occur in them is not directly applicable. Statistical models, such as the vector-space model or variants of the probabilistic model that express documents and queries as Bags-of-Words (BOW) [1] are too poor. Even though fully lexicalized models are well established, in recent years syntactic and semantic structures expressing richer linguistic structures are becoming essential in complex IR tasks, such as Question Classification [21] and Passage Ranking [3] in Question Answering (QA) or Sentiment Analysis Opinion Mining (OM) [12]. The major problem here is that fine-grained phenomena are targeted, and lexical information alone is not sufficient.

The capabilities of the BOW retrieval models do not alway provide a robust solution to these real retrieval needs. For example, in a QA system a BOW IR retrieves documents matching a query, but the QA system actually needs documents that contain answers. The question analysis is thus crucial for the QA system to model the user information needs and to retrieve a proper answer. This is made available when the linguistic and semantic constraints imposed by the question are satisfied by an answer, thus requiring a effective selection of answer-bearing passages.

Language learning systems allow to generalize linguistic observations into rules and patterns as statistical models of higher level semantic inferences. Statistical learning methods make the assumption that lexical or grammatical observations are useful hints for modeling different semantic inferences, such as in document topical classification, predicate and role recognition in sentences as well as question classification in Question Answering. Lexical features here include lemmas, multiword expressions or Named Entities that can be directly observed in the texts. Features are then generalized into predictive components in the final model, induced from the training examples. Obviously, lexical information usually implies different words to provide different contributions but usually neglect other crucial linguistic properties, such as word ordering.

The information about the sentence syntactic structure can be thus exploited and symbolic expressions derived from the parse trees of training examples are used as features for language learning systems. These features denote the position and the relationship between words that can be seemingly realized by different trees independently from irrelevant differences. For example, in a declarative sentence (such as in a `S←NP VP` structure), the relationship between a verbal predicate (`VP`) and its immediately preceding grammatical subject (`NP`) is literally translated in the feature `VP↑VP↑S↓NP`, where arrows indicate upward or downward movements through the tree. Linear kernels over the resulting *Parse Tree Path* features are employed in NLP tasks such as for Semantic Role Labeling [14] or Opinion Mining [22]. This idea is further expanded in tree kernels, introduced by [5]. These model similarity between training examples as a function of the shared subtrees in their corresponding parses. Tree kernels have been successfully applied to different tasks ranging from parsing [5] to semantic role labeling [19]. Tree kernels are known to determine a better grammatical representation for the targeted examples and provide an implicit method for robust feature engineering.

However, the adoption of grammatical features and tree kernels is still affected by significant drawbacks. First, strict requirements exist in terms of the size of the training data set as high dimensionality spaces are generated, whose data sparseness can be prohibitive. Usually, the application of exact learning algorithms gives rise to complex training processes whose convergence is quite slow. Although specific forms of optimization have been proposed to limit their inherent complexity (e.g. [18]), tree kernels do not scale well over very large training data sets. Finally it must be noticed that most of the methods extracting grammatical features from parse trees, are strongly biased by parsing errors.

We want to explore here a possible solution to the above problems through the adoption of shallow but more consistent grammatical features that avoid the use of a full parser in semantic tasks. Parsing accuracy is highly varying across corpora, and it is often poorly effective for some natural languages or application domains where limited resources are available or the syntactic structure of the test instances is very different with respect to the training material. In particular [7] investigates the accuracy loss of well known syntactic parsers applied to micro-blogging datasets. In particular they observed a drastic drop in performance moving from the in-domain test set to the new Twitter dataset. Avoiding the adoption of full parsing obviously increases the number and nature of possible uses of language technologies in a variety of complex NLP applications. In IR, part of speech information has been generally used for stemming, generating stop-word lists, and identifying pertinent terms or phrases in documents and/or in queries. Generally, the state of the art in IR systems tend to benefit from the adoption of parts of speech to index or retrieve information [24].

The open research questions are: which shallow grammatical representation is suitable to support the learning of fine-grained semantic models? Which grammatical generalizations can be usefully achieved over shallow syntactic representations for sentence-based inferences?

In the rest of this work, we show how embedding shallow grammatical information in a sentence representation, as a special case of enriched lexical information, produces useful generalizations in standard machine learning settings. Empirical findings in support to this thesis are discussed against two complex sentence-based semantic tasks, i.e. question classification and sentiment analysis in micro-blogging.

2 Shallow Parsing and Grammatical Feature engineering

Grammatical feature engineering is required as lexical information alone is, in general, not sufficient to characterize linguistic generalizations useful for fine-grained semantic inferences. For example, sentence (3) is the appropriate answer for the question (1), although both sentences (2) and (3) are reasonable candidates.

What ***French province is Cognac produced in****?* (1)

The grapes which ***produce*** *the* ***Cognac*** *grow* ***in*** *the* ***province*** *and the* ***French*** *government ...* (2)

Cognac is *a brandy* ***produced in*** *Poitou-Charentes.* (3)

Suppose we use a lexical overlap rule for a Question Answering (QA) task: given the overlapping terms outlined in bold[1], it would result in the wrong answer (2). A simple lexical overlap model is too simplistic, as syntactic information characterizing the individual sentences (1) and (3) is here necessary. Syntactic features provide more information to estimate the similarity between the question and the candidate answers, as in general explored by tree kernels in Answer Classification/Re-ranking [20]. The parse tree in Figure 1 corresponds to sentence (3) and represents:

- *lexical information* through its terminal nodes (e.g., words as $Cognac$, is, $\ldots$)
- *Coarse-grained grammatical information* through the POS tag characterizing preterminal nodes (e.g. NNP or VBZ)
- *Fine-grained grammatical information* as subtrees correspond to the production rules of the underlying *context free grammar* (CFG).

Examples of the CFG rules involved in Figure 1 are: $S \rightarrow NP\ VP$, $VP \rightarrow VBZ\ NP$, $NP \rightarrow NPP$ or $NP \rightarrow DT\ NN$. Stochastic context free grammars (e.g. [4]), are generative models for parse trees, seen as complex joint events, whose overall probability depends on the individual CFG rules (i.e., subtrees), and lexical information as well. Our aim here is to acquire these rules implicitly, as a side effect of the learning for semantic inference process. Specific features can in fact be designed to surrogate the syntactic structures of the parse tree, implicitly. Observable POS tag sequences correspond to subtrees and can be considered their shallow counterpart.

[1] Sentence (2) shares five terms with the sentence (1), while (3) shares only four terms.

They express linearly special properties, in analogy with the Parse Tree Paths in [9]. In other words, subtrees can be artificially replaced introducing POS tag sequences (or POS n-grams), instead of parse tree fragments. The idea is that the syntactic structure of a sentence could be surrogated as the POS n-grams, instead of the set of possible syntactic tree fragments, as used by tree kernels. For example, the partial tree expressed by `VP`→`VBN PP` in Fig. 1 can be represented through the pseudo token given by *VBN-IN-NNP*.

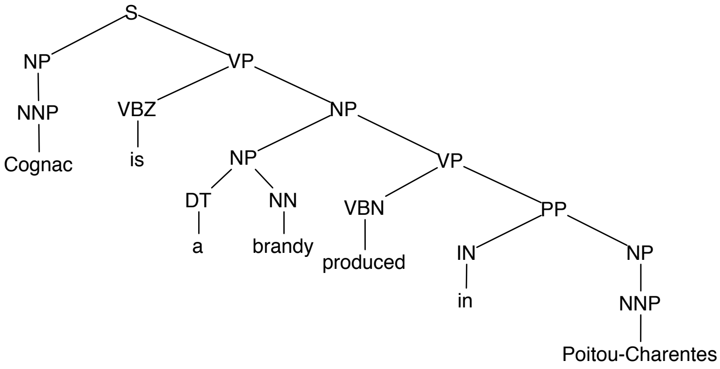

Fig. 1. Example of parse tree associated to sentence (3)

Lexicalized features (i.e., true words) as well as shallow syntactic information (i.e., the POS n-grams) are thus made available as flat features, thus constraining the capacity of the underlying learning machine. A sentence s of length $|s|$ is thus represented as a set of words (in a bag-of-word fashion), extended by the pseudo tokens defining the corresponding POS tag sequences whose length is smaller that n (n-POS tag grams). Given the word sequence $s = \{w_1, \ldots, w_{|s|}\}$ whose corresponding part-of-speeches are $\{pos_1, \ldots, pos_{|s|}\}$, the representation of the pseudo tokens is the set of pairs $\{(w_1.pos_1), \ldots, (w_{|s|}.pos_{|s|}\}$, where each lemmatized word is coupled with its POS tag.

Moreover, in order to capture syntactic structures of interest, POS tags are also mapped into pseudo-tokens expressing their sequences (i.e., POS n-grams). Given n as the maximal size of the extracted sequences, every subsequence of length at most n is mapped into a pseudo-token. These novel *grammatical* tokens of length Δ are expressed as $\{p_j, \ldots, p_{j+\Delta}\}$ where $\Delta = 1, ..., n$. In these patterns the representation of prepositions (POS tag *IN*) is made explicit. Every position $k \in [j, j + \Delta]$ for which pos_k=*IN* is represented through w_k itself, so that *at-NP* or *of-DT-NN* are obtained as pseudo-tokens for fragments such as "*at Whitlock*" or "*of the vineyard*". The representation of sentence (3) is shown in Table 2, where words $(w_i.pos_i)$ and n-gram tokens are shown.

Table 1. Representation of lexical and grammatical information for sentence (3)

unigrams	cognac.NNP be.VBZ a.DT brandy.NN produce.VBN in.IN poitou-charentes.NNP
2-grams	NNP-VBZ VBZ-DT DT-NN NN-VBN VBN-in in-NNP NNP-.
3-grams	NNP-VBZ-DT VBZ-DT-NN DT-NN-VBN NN-VBN-in VBN-in-NNP in-NNP-.
4-grams	NNP-VBZ-DT-NN VBZ-DT-NN-VBN DT-NN-VBN-in NN-VBN-in-NNP VBN-in-NNP-.

2.1 Shallow Syntactic Features for Question Classification

In Question Answering three main processing stages are foreseen: question processing, document retrieval and answer extraction [16]. Question processing is usually centered around the so called *question classification* (*QC*) task that maps a question into one of k predefined answer classes [17]. Typical examples of classes characterize different answer strategies and range from questions regarding *persons* or *organizations* (e.g. *Who killed JFK*?) to *definition* questions (e.g. *What is a perceptron*?) or *modalities* (e.g. *How fast does boiling water cool?*). Highly accurate *QC* systems apply supervised machine learning techniques, e.g. Support Vector Machines (SVMs) [20, 23] or the SNoW model [17], where questions are encoded using a variety of lexical, syntactic and semantic features. In [17], it has been shown that the questions' syntactic structure contributes remarkably to the classification accuracy. This task is thus strictly syntax-dependent, especially because individual sentences are targeted.

As questions can be regarded as individual sentences, we will adopt the feature extraction scheme proposed in Table 2 for our QC models. These features represent both lexical and grammatical information that can be efficiently feed a statistical classifier based on linear kernels. Section 3.1 will discuss comparative experiments with previous works on Question Classification.

2.2 Shallow Syntactic Features for Sentiment Analysis over micro-blogging

Microblogging has been already established as a significant form of electronic word-of-mouth for sharing opinions, suggestions and consumer reviews concerning ideas, products or brands. Microblogging is also referred to as micro-sharing or *Twittering* (from Twitter[2] by far the most popular microblogging application). While opinion mining over traditional text sources (e.g. movie reviews or forums) has been significantly studied [22], sentiment analysis over tweets has a more recent history, [10] or [2]. It has been usually addressed on the basis of only lexical information whereas the syntactic structure of tweets is often neglected [22]. In [25] the linguistic redundancy in Twitter is investigated and several types of linguistic features are tested in a supervised setting, showing that tweet syntactic structure does not provide alone a statistically significant contribution with respect to lexical typed features. The main problem of syntax-driven

[2] http://www.twitter.com

approaches over tweets is the quality of the available grammatical information as tweets are sentences lacking of a proper grammatical structure.

Here the modeling through POS n-grams is suitable to overcome these problems, as it provides a simpler representation of the tweets' syntax and, on the other hand, it should be more robust as for tagging accuracy. However even POS taggers, trained over standard texts, may be inadequate, as the linguistic form of tweets is rather non standard with a large use of jargon and shortcuts. An interesting finding in [7] was that one of the main cause of the syntactic parsing errors over the Twitter dataset is due to the propagation of part-of-speech tagging errors. In line with other works (see for example [10] or [15]), we propose to pre-process tweets before a *standard* POS tagger is applied. This avoids the noise in applying traditional POS tagging to odd symbols (e.g. re-tweets or emoticons) or jargon expressions and also reduces data sparseness, as canonical forms are adopted. The following set of actions is applied before training:

- fully capitalized words are first converted in their lowercase counterpart, i.e. "*DOG*" into "*dog*", before applying POS tagging
- reply marks (i.e. @*user_name*) are replaced with the pseudo-token `USER` whose POS tag is set back to `PUSER` after POS tagging
- hyperlinks are replaced by the token `LINK` whose POS is `PLINK`
- hash tags (i.e. *#thread_name*) are replaced by the pseudo-token `THREAD` whose POS is imposed to `PTHREAD`
- repeated letters and punctuation characters (e.g. *looove*, *loooove* or *!!!*) are cleansed as they cause high levels of lexical data sparseness. Characters occurring more than twice are all replaced with a double occurrence expression, so that *looove* or *!!!* are mapped into *loove* or *!!*, respectively
- all emoticons, e.g. :-) or *:P*, are used as sentence separators although they are systematically misinterpreted by a standard POS tagger. Accordingly, they are first replaced with a full stop "." and then recovered at their original form after POS tagging. Their POS is always set to `SMILE`.

After the above pre-processing phase, a tweet like *@jdoe I looove Twitter! :-) http://twitpic.com/2y2e0* can be represented according to the model proposed in Section 2. Here the lists of lexical unigrams and grammatical n-grams are reported:

```
USER.PUSER i.PRP loove.VBP twitter.NNP!.PUNC :-).SMILE LINK.PLINK
PUSER_PRP PRP_VBP VBP_NNP NNP_PUNC PUNC_SMILE SMILE_PLINK USER_PRP_VBP ...
```

As it is clear from the example, the resulting POS sequences are able to better capture the intended syntax and act as good models of relevant grammatical relations: the sequence `USER.PUSER i.PRP loove.VBP` ..., for example, is a good hint for the positive bias introduced by $loove$ as a verb.

3 Performance Evaluation

In this section we evaluate the use of POS n-grams in two applications previously discussed as standard example of different semantic inferences useful for IR. In all the experiments POS tagging is carried out by the tagger available in the LTH parser [13]. The performance achievable by POS n-grams is thus compared with the one derived by richer grammatical representations based on parse trees.

3.1 Question Classification Results

This first experiment studies the impact of combining lexical and shallow syntactic information (i.e. POS n-grams), on question classification. The targeted dataset is the UIUC corpus, largely adopted for benchmarking [17]. UIUC contains a training set of 5,452 questions and a test set of 500 questions, both extracted from TREC. Question classes are organized in two levels of granularity. At the first level, 6 coarse-grained classes are defined, like ABBREVIATION, ENTITY, DESCRIPTION. A second level explodes the first level classes into a set of 50 fine-grained sub-classes, e.g., *Plant* and *Food* are subclasses of the ENTITY category.

SVM learning is applied over the feature vectors discussed in Section 2.1 and multi-classification is modeled through a *one-vs-all* scheme. The quality of classification is measured through accuracy, i.e. the percentage of questions associated with the correct class. A development set is derived from the 20% of the training material. In the experiments two sentence models are compared:

- *POS tagged Unigrams (PU)*: a question is mapped into a bag of POS tagged lemmas, i.e. into pairs of $(lemma.pos)$. This model is based only on lexical information.
- *POS n-grams (PnG)*: each question is modeled by augmenting the PU model through the shallow syntactic information provided by the sequence of n-grams of POS tags, with $n < 4$. The POS of Wh-determiners and prepositions are replaced in the individual POS n-grams by the corresponding lemmas.

In this evaluation the voted perceptron [8] and SMV^{light} [11] have been both applied[3] . Results, compared with the results achieved by the system discussed in [23] on the same UIUC dataset, are shown in Table 2. The authors combine a kernel classifier based on BOW with two semantic kernels: one (i.e. K(LS)) is based on Latent Semantic Indexing applied to Wikipedia, and the other (i.e. K(semRel)) uses semantic information acquired through *manually constructed lists of words*, i.e., a task-specific lexicon related to the answer types.

In the coarse-grained test, i.e. the question classification with respect to the 6 coarse grained classes, Table 2 shows how the syntactic generalization supported by the PnG model achieves the best known results on the UIUC dataset, i.e., 91.8% that correspond to the accuracy reported by a tree kernel approach [20], without any semantic extension. This improves the best results of [23] (i.e., the $K(BOW) + K(LS) + K(semRel)$) that refer to a *task-dependent use of manually annotated resources*. Note how the kernel $K(LS)$ that uses only lexical information, gathered by an external corpus like Wikipedia [23] is also weaker than the PnG model, that makes no use of trees or other

[3] In the experiments a polynomial kernel of degree 2 has been applied with SMV^{light}, as it achieved the best result on the development set

Table 2. Accuracy measures for the QC task

Kernel	Coarse Task	Fine-grain Task
PU (VotedPerc)	89.2%	81.4%
PU (SVM)	89.4%	83.8%
PnG (VotedPerc)	91.4%	84.0%
PnG (SVM)	**91.8%**	84.8%
[23]		
K(BOW)	86.4%	80.8%
K(LS)	70.4%	71.2%
K(BOW)+K(LS)	90.0%	83.2%
K(BOW)+K(LS)+K(semRel)	90.8%	**85.6%**
[20]		
Tree Kernels K(BOW)+K(*PartialTrees*)	**91.8%**	-

resources. The results in Table 2 are also remarkable from a computational point of view: the PnG method only requires POS tagged sentences and no parsing. Moreover, the training time of tree kernel based SVMs on benchmarking data sets are in the order of hours or days for large training collections (e.g., Prop Bank, as reported in [18]).

In [6] an extension to the tree kernel formulation has been proposed, i.e. the semantic Smoothed Partial Tree Kernel that enriches the similarity among syntactic tree structures with lexical information gathered by en external corpus, in line with the K(LS) described in [23]. State-of-the art results of 94.8% have been obtained in the coarse-grained test. However it is still a complex approach that need explicit syntactic parsing of the sentences and an external corpus that provides lexical knowledge. This is beyond the scope of this work, that aims at providing an efficient and practical engineering method for natural language learning systems. The training complexity of the proposed models is very low. Consider that for a short sentence (i.e. a question or a micro-blogging message) the number of feature is reduced. For example a sentence of 10 words, will generate 10 lexical, 9 bi-gram, 8 three-gram and 7 four-gram features, i.e. a feature vector of 34 features. It generates a hi-dimensional but very sparse space, where both SVM and the vote perceptron algorithms can very effectively find a solution. The efficiency of the proposed method in the QC task is thus proved, as the *PnG* model has been trained over 5,452 examples in less than 2 minutes and 40 seconds, with SMV^{light} and the voted perceptron, respectively.

3.2 Sentiment Analysis Results

The POS n-grams model has been also applied in the task of Sentiment Analysis over tweets, as introduced in Section 2.2. The goal here is to classify a tweet according to its sentiment polarity. The adopted dataset is Twitter Sentiment, released by [10][4], as other studies (e.g. [2]) do not allow a full comparative analysis. It provides a training

[4] `http://www.stanford.edu/∼alecmgo/cs224n/twitterdata.2009.05.25.c.zip`

set automatically generated by selecting the positive (or negative) examples from the tweets containing positive (or negative) emoticons, e.g. :-) (or :-( ). The test set, also made available by [10], includes 183 tweets, manually annotated according to their binary sentiment polarity, i.e. ± 1. Each tweet is modeled as a feature vector, including words as well as the pseudo-tokens generated in the pre-processing phase, including the resulting POS n-grams (see Section 2.2). SMV^{light} has been applied, with a 50-50% train-development splitting: in this setting a linear kernel provided the best results.

Table 3. Experimental results for the Sentiment Analysis task

Unigrams	77.60%
POS tagged Unigrams	82.51%
Noisy POS 4-grams (no pre-proc.)	77.59%
POS 4-grams	**83.61%**
Unigrams [10]	82.20%
POS tagged Unigrams [10]	83.00%

As Table 3 suggests, the results improve on [10], as the adopted grammatical information is helpful. The test set employed in our experiments is slightly more complex, as the *Unigrams* model achieves a significantly worse result than in [10]. Moreover, without pre-processing, POS tags are inaccurate and this reflects in the lower performances of the Noisy POS 4-grams model. Our approach achieves a new state-of-art (i.e. 83.61%) on the dataset. This results due to the grammatical information provided by the POS n-grams and the contribution of the proposed pre-processing method is crucial. When no pre-processing is applied, the noise introduce by the POS-tagger would produce a consistent performance reduction, i.e. 77.59% vs 82.51%. Error analysis

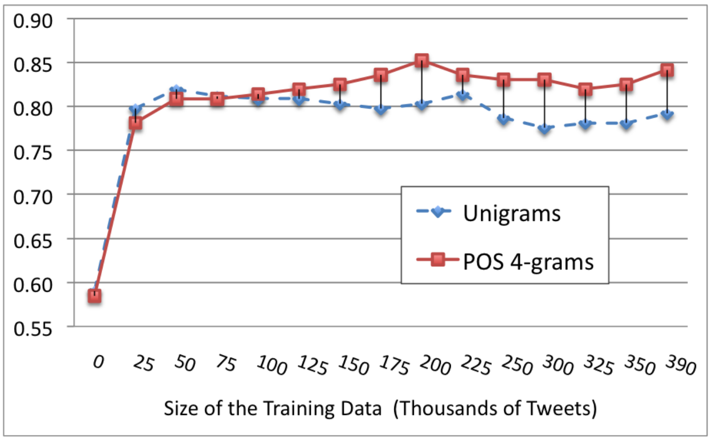

Fig. 2. Twitter Sentiment Analysis: accuracy

suggests that mistakes (e.g. the positive polarity given to the tweet ”*Kobe is the best in the world not Lebron*”) are due to lack of information. If LeBron James (and not Kobe) is the focus then the polarity is negative. But the alternative decision would have been perfectly acceptable, otherwise. Figure 2 reports the learning curve for the system with and without POS n-grams: POS n-grams are responsible of a faster convergence to higher accuracy levels.

4 Conclusions

In this paper shallow grammatical features as sequences of POS tags (i.e. POS n-grams) are proposed as a robust and effective model of grammatical information in different semantic tasks. Every experiment shows that state-of-the-art results are achieved or closely approximated by our modeling. Although standard training algorithms are here adopted, simple kernels over POS n-grams are quite effective, as for example the sentiment analysis tests demonstrate. Surprisingly, in Question Classification our model equals the accuracy of a performant tree kernel. The training complexity of the proposed models is very low. Although several optimization methods for tree kernel learning have been proposed (e.g. [6, 18]), our simpler approach is more applicable by posing much weaker requirements in terms of quality and size of the annotated datasets. This makes the proposed technology quite appealing for complex NLP and IR applications, such as the treatment of noisy sources that current micro-blogging trends require. This is also shown by the performances observed in the tweet sentiment analysis task, for which state-of-the-art results are obtained.

References

1. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)
2. Barbosa, L., Feng, J.: Robust sentiment detection on twitter from biased and noisy data. In: Coling 2010: Posters. pp. 36–44. Coling 2010 Organizing Committee, Beijing, China (August 2010)
3. Bilotti, M.W., Elsas, J.L., Carbonell, J., Nyberg, E.: Rank learning for factoid question answering with linguistic and semantic constraints. In: Proceedings of ACM CIKM (2010)
4. Collins, M.: Three generative, lexicalised models for statistical parsing. In: Proceedings of ACL 1997. pp. 16–23 (1997)
5. Collins, M., Duffy, N.: Convolution kernels for natural language. In: Proceedings of Neural Information Processing Systems (NIPS). pp. 625–632 (2001)
6. Croce, D., Moschitti, A., Basili, R.: Structured lexical similarity via convolution kernels on dependency trees. In: Proceedings of EMNLP. Edinburgh, Scotland, UK. (2011)
7. Foster, J., Özlem Çetinoğlu, Wagner, J., Roux, J.L., Hogan, S., Nivre, J., Hogan, D., van Genabith, J.: #hardtoparse: Pos tagging and parsing the twitterverse. In: Prooceedings of AAAI-11 Workshop on Analysing Microtext. San Francisco, CA (August 2011)
8. Freund, Y., Schapire, R.E.: Large margin classification using the perceptron algorithm. Machine Learning Journal 37(3), 277–296 (1999)
9. Gildea, D., Jurafsky, D.: Automatic Labeling of Semantic Roles. Computational Linguistics 28(3), 245–288 (2002)

10. Go, A., Bhayani, R., Huang, L.: Twitter Sentiment Classification using Distant Supervision. In: CS224N Project Report, Stanford (2009)
11. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: In Proceedings of the European Conference on Machine Learning (1998)
12. Johansson, R., Moschitti, A.: Extracting opinion expressions and their polarities – exploration of pipelines and joint models. In: Proceedings of ACL-HLT. Portland, Oregon, USA (2011)
13. Johansson, R., Nugues, P.: Dependency-based syntactic-semantic analysis with propbank and nombank. In: Proceedings of CoNLL-2008. Manchester, UK (August 16-17 2008)
14. Johansson, R., Nugues, P.: The effect of syntactic representation on semantic role labeling. In: Proceedings of COLING. Manchester, UK (August 18-22 2008)
15. Kaufmann, J., Kalita, J.: Syntactic normalization of twitter messages. In: International Conference on Natural Language Processing (2010)
16. Kwok, C.C.T., Etzioni, O., Weld, D.S.: Scaling question answering to the web. In: WWW. pp. 150–161 (2001)
17. Li, X., Roth, D.: Learning question classifiers. In: Proceedings of ACL'02 (2002)
18. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: ECML. pp. 318–329. Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Proceedings, Berlin, Germany (September 2006)
19. Moschitti, A., Pighin, D., Basili, R.: Tree kernels for semantic role labeling. Computational Linguistics 34 (2008)
20. Moschitti, A., Quarteroni, S., Basili, R., Manandhar, S.: Exploiting syntactic and shallow semantic kernels for question answer classification. In: In Proc. of ACL-07. pp. 776–783 (2007)
21. Moschitti, A., Quarteroni, S., Basili, R., Manandhar, S.: Exploiting syntactic and shallow semantic kernels for question/answer classification. In: Proceedings of ACL'07 (2007)
22. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (Jan 2008)
23. Tomάs, D., Giuliano, C.: A semi-supervised approach to question classification. In: Proceedings of the 17th European Symposium on Artificial Neural Networks, Bruges, Belgium (2009)
24. Voorhees, E.M., Harman, D.: Overview of the seventh text retrieval conference trec-7. In: Proceedings of the Seventh Text REtrieval Conference (TREC-7. pp. 1–24 (1998)
25. Zanzotto, F.M., Pennacchiotti, M., Tsioutsiouliklis, K.: Linguistic redundancy in twitter. In: Proceedings of 2011 Conference on Empirical Methods on Natural Language Processing (EmNLP) (2011)