Semantic Kernels for Semantic Parsing

Iman Saleh
Faculty of Computers and Information
Cairo University
iman.saleh@fci-cu.edu.eg

Alessandro Moschitti, Preslav Nakov, Lluís Màrquez, Shafiq Joty
ALT Research Group
Qatar Computing Research Institute
{amoschitti,pnakov,lmarquez,sjoty}@qf.org.qa

Abstract

We present an empirical study on the use of semantic information for Concept Segmentation and Labeling (CSL), which is an important step for semantic parsing. We represent the alternative analyses output by a state-of-the-art CSL parser with tree structures, which we rerank with a classifier trained on two types of semantic tree kernels: one processing structures built with words, concepts and Brown clusters, and another one using semantic similarity among the words composing the structure. The results on a corpus from the restaurant domain show that our semantic kernels exploiting similarity measures outperform state-of-the-art rerankers.

1 Introduction

Spoken Language Understanding aims to interpret user utterances and to convert them to logical forms or, equivalently, to database queries, which can then be used to satisfy the user's information needs. This process is known as Concept Segmentation and Labeling (CSL), also called *semantic parsing* in the speech community: it maps utterances into meaning representations based on semantic constituents. The latter are basically word sequences, often referred to as concepts, attributes or semantic tags. CSL makes it easy to convert spoken questions such as "cheap lebanese restaurants in doha with take out" into database queries.

First, a language-specific semantic parser tokenizes, segments and labels the question:

[*Price* cheap] [*Cuisine* lebanese] [*Other* restaurants in] [*City* doha] [*Other* with] [*Amenity* take out]

Then, label-specific normalizers are applied to the segments, with the option to possibly relabel mislabeled segments:

[*Price* low] [*Cuisine* lebanese] [*City* doha] [*Amenity* carry out]

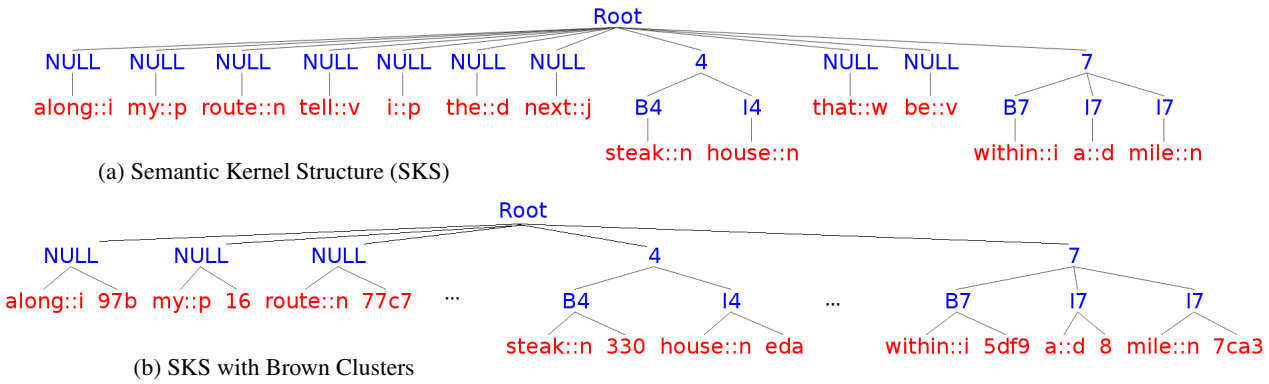
Finally, a database query is formed from the list of labels and values, and is then executed against the database, e.g., MongoDB; a backoff mechanism may be used if the query has not succeeded.

```
{$and [{cuisine:"lebanese"},{city:"doha"},
  {price:"low"},{amenity:"carry out"}]}
```

The state-of-the-art of CSL is represented by conditional models for sequence labeling such as Conditional Random Fields (CRFs) (Lafferty et al., 2001) trained with simple morphological and lexical features. The basic CRF model was improved by means of reranking (Moschitti et al., 2006; Dinarelli et al., 2012) using structural kernels (Moschitti, 2006). Although these methods exploited sentence structure, they did not use syntax at all. More recently, we applied shallow syntactic structures and discourse parsing with slightly better results (Saleh et al., 2014). However, the most obvious models for semantic parsing, i.e., rerankers based on semantic structural kernels (Bloehdorn and Moschitti, 2007b), had not been applied to semantic structures yet.

In this paper, we study the impact of semantic information conveyed by Brown Clusters (BCs) (Brown et al., 1992) and semantic similarity, while also combining them with innovative features. We use reranking, similarly to (Saleh et al., 2014), to select the best hypothesis annotated with concepts predicted by a local model. The competing hypotheses are represented as innovative trees enriched with the semantic concepts and BC labels. The trees can capture dependencies between sentence constituents, concepts and BCs. However, extracting explicit features from them is rather difficult as their number is exponentially large. Thus, we rely on (*i*) Support Vector Machines (Joachims, 1999) to train the reranking functions and on (*ii*) structural kernels (Moschitti, 2010; Moschitti, 2012; Moschitti, 2013) to automatically encode tree fragments that represent syntactic and semantic dependencies from words and concepts.

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 436–442,
October 25-29, 2014, Doha, Qatar.



(a) Semantic Kernel Structure (SKS)

(b) SKS with Brown Clusters

Figure 1: CSL structures: standard and with Brown Clusters.

We further apply a semantic kernel (SK), namely the Smoothed Partial Tree Kernel (Croce et al., 2011), which uses the lexical similarity between the tree nodes, while computing the substructure space. This is the first time that SKs are applied to reranking hypotheses. This (*i*) makes the global sentence structure along with concepts available to the learning algorithm, and (*ii*) enables computing the similarity between lexicals in syntactic patterns that are enriched by concepts.

We tested our models on the *Restaurant* domain. Our results show that: *(i)* The basic CRF parser, which uses semi-Markov CRF, or semi-CRF (Sarawagi and Cohen, 2004), is already very accurate; it achieves F_1 scores over 83%, making any further improvement very hard. (*ii*) The upper-bound performance of the reranker is very high as well, i.e., the correct annotation is generated in the list of the first 100 hypotheses in 98.72% of the cases. (*iii*) SKs significantly improve over the semi-CRF baseline and our previous state-of-the-art reranker exploiting shallow syntactic patterns (Saleh et al., 2014), as shown by extensive comparisons using several systems. (*iv*) Making BCs effective requires a deeper study.

2 Related Work

One of the early approaches to CSL was that of Pieraccini et al. (1991), where the word sequences and *concepts* were modeled using Hidden Markov Models (HMMs) as observations and hidden states, respectively. Generative models were exploited by Seneff (1989) and Miller et al. (1994), who used stochastic grammars for CSL. Other discriminative models followed such preliminary work, e.g., (Rubinstein and Hastie, 1997; Santafé et al., 2007; Raymond and Riccardi, 2007). CRF-based models are considered to be the state of the art in CSL (De Mori et al., 2008).

Another relevant line of research are the semantic kernels, i.e., kernels that use lexical similarity between features. One of the first that applyed LSA was (Cristianini et al., 2002), whereas (Bloehdorn et al., 2006; Basili et al., 2006) used WordNet. Semantic structural kernels of the type we use in this paper were first introduced in (Bloehdorn and Moschitti, 2007a; Bloehdorn and Moschitti, 2007b). The most advanced model based on tree kernels, which we also use in this paper, is the Smoothed PTK (Croce et al., 2011).

3 Reranking for CSL

Reranking is applied to a list of N annotation hypotheses, which are generated and sorted by the probability to be globally correct as estimated using local classifiers or global classifiers that only use local features. Then, a reranker, typically a meta-classifier, tries to select the best hypothesis from the list. The reranker can exploit global information, and specifically, the dependencies between the different concepts, which are made available by the local model. We use semi-CRFs for the local model as they yield the highest accuracy in CSL (when using a single model) and preference reranking for the global reranker.

3.1 Preference Reranking (PR)

PR uses a classifier $\mathcal{C}$, which takes a pair of hypotheses $\langle H_i, H_j \rangle$ and decides whether H_i is better than H_j. Given a training question Q, positive and negative examples are built for training the classifier. Let H_1 be the hypothesis with the lowest error rate with respect to the gold standard among all hypotheses generated for question Q. We adopt the following approach for example generation: the pairs $\langle H_1, H_i \rangle$ $(i = 2, 3, \ldots, N)$ are positive examples, while $\langle H_i, H_1 \rangle$ are considered negative.

At testing time, given a new question Q′, $\mathcal{C}$ classifies all pairs $\langle H_i, H_j \rangle$ generated from the annotation hypotheses of Q′: a positive classification is a vote for H_i, otherwise the vote is for H_j, where the classifier score can be used as a weighted vote. H_k are then ranked according to the number (sum) of the votes (weighted by score) they receive.

We build our reranker with SVMs using the following kernel: $K(\langle H_1, H_2 \rangle, \langle H'_1, H'_2 \rangle) = \phi(\langle H_1, H_2 \rangle) \cdot \phi(\langle H'_1, H'_2 \rangle) \triangleq \big(\phi(H_1) - \phi(H_2)\big) \cdot \big(\phi(H'_1) - \phi(H'_2)\big) = \phi(H_1)\phi(H'_1) + \phi(H_2)\phi(H'_2) - \phi(H_1)\phi(H'_2) - \phi(H_2)\phi(H'_1) = S(H_1, H'_1) + S(H_2, H'_2) - S(H_1, H'_2) - S(H_2, H'_1)$. We consider H as a tuple $\langle T, \vec{v} \rangle$ composed of a tree T and a feature vector $\vec{v}$. Then, we define $S(H, H') = S_{\text{TK}}(T, T') + S_{\text{v}}(\vec{v}, \vec{v}')$, where S_{TK} computes one of the tree kernel functions defined in 3.2 and 3.3; and S_{v} is a kernel (see 3.4), e.g., linear, polynomial, Gaussian, etc.

3.2 Tree kernels (TKs)

TKs measure the similarity between two structures in terms of the number of substructures they share. We use two types of tree kernels: (*i*) Partial Tree Kernel (PTK), which can be effectively applied to both constituency and dependency parse trees (Moschitti, 2006). It generates all possible connected tree fragments, e.g., sibling nodes can be also separated and can be part of different tree fragments: a fragment is any possible tree path, and other tree paths are allowed to depart from its nodes. Thus, it can generate a very rich feature space. (*ii*) The smoothed PTK or semantic kernel (SK) (Croce et al., 2011), which extends PTK by allowing soft matching (i.e., via similarity computation) between nodes associated with different but related lexical items. The node similarity can be derived from manually annotated resources, e.g., WordNet or Wikipedia, as well as using corpus-based clustering approaches, e.g., latent semantic analysis (LSA), as we do in this paper.

3.3 Semantic structures

Tree kernels allow us to compute structural similarities between two trees; thus, we engineered a special structure for the CSL task. In order to capture the structural dependencies between the semantic tags,[1] we use a basic tree (see for example Figure 1a), where the words of a sentence are tagged with their semantic tags.

More specifically, the words in the sentence constitute the leaves of the tree, which are in turn connected to the pre-terminals containing the semantic tags in BIO notation ('B'=*begin*, 'I'=*inside*, 'O'=*outside*). The BIO tags are then generalized in the upper level, and joined to the Root node. Additionally, part-of-speech (POS) tags[2] are added to each word by concatenating it with the string "::L", where L is the first letter of the POS-tags of the words, e.g., *along*, *my* and *route*, receive i, p and n, which are the first letters of the POS-tags IN, PRN and NN, respectively. SK applied to the above structure can generate powerful semantic patterns such as [Root [4-Cuisine [similar_to(*stake house*)]][7-Loc [similar_to(*within a mile*)]]], e.g., for correctly labeling new clauses like *Pizza Parlor in three kilometers*. The BC labels, represented as cluster IDs, are simply added as siblings of words as shown in Fig. 1b.

3.4 Feature Vectors

For the sake of comparison, we also devoted some effort towards engineering a set of features to be used in a flat feature-vector representation. These features can be used in isolation to learn the reranking function, or in combination with the kernel-based approach (as a composite kernel using a linear combination). They belong to the following four categories: (*i*) CRF-based: these include the basic features used to train the initial semi-CRF model; (*ii*) n-gram based: we collected 3- and 4-grams of the output label sequence at the level of concepts, with artificial tags inserted to identify the start ('S') and end ('E') of the sequence.[3] (*iii*) Probability-based, computing the probability of the label sequence as an average of the probabilities at the word level in the N-best list; and (*iv*) DB-based: a single feature encoding the number of results returned from the database when constructing a query using the conjunction of all semantic segments in the hypothesis.

4 Experiments

The experiments aim at investigating the role of feature vectors, PTK, SK and BCs in reranking. We first describe the experimental setting and then we move into the analysis of the results.

[1] They are associated with the following IDs: 0-Other, 1-Rating, 2-Restaurant, 3-Amenity, 4-Cuisine, 5-Dish, 6-Hours, 7-Location, and 8-Price.

[2] We use the Stanford tagger (Toutanova et al., 2003).

[3] For instance, if the output sequence is *Other-Rating-Other-Amenity* the 3-gram patterns would be: S-*Other-Rating, Other-Rating-Other, Rating-Other-Amenity*, and *Other-Amenity*-E.

	Train	Devel.	Test	Total
semi-CRF	6,922	739	1,521	9,182
Reranker	7,000	3,695	7,605	39,782

Table 1: Number of instances and pairs used to train the semi-CRF and rerankers, respectively.

4.1 Experimental setup

Dataset. In our experiments, we used questions annotated with semantic tags, which were collected through crowdsourcing on Amazon Mechanical Turk and made available[4] by McGraw et al. (2012). We split the dataset into training, development and test sets. Table 1 shows the number of examples and example pairs we used for the semi-CRF and the reranker, respectively. We subsequently split the training data randomly into 10 folds. We used cross-validation, i.e., iteratively training with 9 folds and annotating the remaining fold, in order to generate the N-best lists of hypotheses for the entire training dataset. We computed the 100-best hypotheses for each example. We then used the development dataset to test and tune the hyper-parameters of our reranking model. The results on the development set, which we will present in Section 4.2 below, were obtained using semi-CRF and reranking models trained on the training set.

Data representation. Each hypothesis is represented by a semantic tree, a feature vector (explained in Section 3), and two extra features: (*i*) the semi-CRF probability of the hypothesis, and (*ii*) its reciprocal rank in the N-best list.

Learning algorithm. We used the SVM-Light-TK[5] to train the reranker with a combination of tree kernels and feature vectors (Moschitti, 2006; Joachims, 1999). We used the default parameters and a linear kernel for the feature vectors. As a baseline, we picked the best-scoring hypothesis in the list, i.e., the output by the regular semi-CRF parser. The setting is exactly the same as that described in (Saleh et al., 2014).

Evaluation measure. In all experiments, we used the harmonic mean of precision and recall (F_1) (van Rijsbergen, 1979), computed at the token level and micro-averaged across the different semantic types.[6]

[4]http://groups.csail.mit.edu/sls/downloads/restaurant/

[5]http://disi.unitn.it/moschitti/Tree-Kernel.htm

[6]We do not consider 'Other' to be a semantic type; thus, we did not include it in the F_1 calculation.

N	1	2	**5**	10	100
F_1	83.03	87.76	**92.63**	95.23	98.72

Table 2: Oracle F_1 score for N-best lists.

Brown Clusters. Clustering groups of similar words together provides a way of generalizing them. In this work, we explore the use of Brown clusters (Brown et al., 1992) in both feature vectors and tree kernels. The Brown clustering algorithm uses an n-gram class model. It first assigns each word to a distinct cluster, and then it merges different clusters in a bottom-up fashion. The merge step is done in a way that minimizes the loss in average mutual information between clusters. The outcome is hierarchical clustering, which we use in our reranking algorithm. To create the Brown clusters, we used the Yelp dataset of reviews.[7] It contains 335,022 reviews about 15,585 businesses; 5,575 of the businesses and 233,839 of the reviews are restaurant-related. This dataset is very similar to the dataset of queries about restaurants we use in our experiments.

Similarity matrix for SK. We compute the lexical similarity for SK by applying LSA (Furnas et al., 1988) to Tripadvisor data. The dataset and the exact procedure for creating the LSA matrix are described in (Castellucci et al., 2013; Croce and Previtali, 2010).

4.2 Results

Oracle accuracy. Table 2 shows the oracle F_1 score for N-best lists of different lengths, i.e., the F_1 that is achieved by picking the best candidate in the N-best list for various values of N. Considering 5-best lists yields an increase in oracle F_1 of almost ten absolute points. Going up to 10-best lists only adds 2.5 extra F_1 points. The complete 100-best lists add 3.5 extra F_1 points, for a total of 98.72. This very high value is explained by the fact that often the total number of different annotations for a given question is smaller than 100. In our experiments, we will focus on 5-best lists.

Baseline accuracy. We computed F_1 for the semi-CRF model on both the development and the test sets, obtaining 83.86 and 83.03, respectively.

Learning Curves. The semantic information in terms of BCs or semantic similarity derived by LSA can have a major impact in case of data scarcity. Therefore, we trained our reranking models with increasing sizes of training data.

[7]http://www.yelp.com/dataset_challenge/

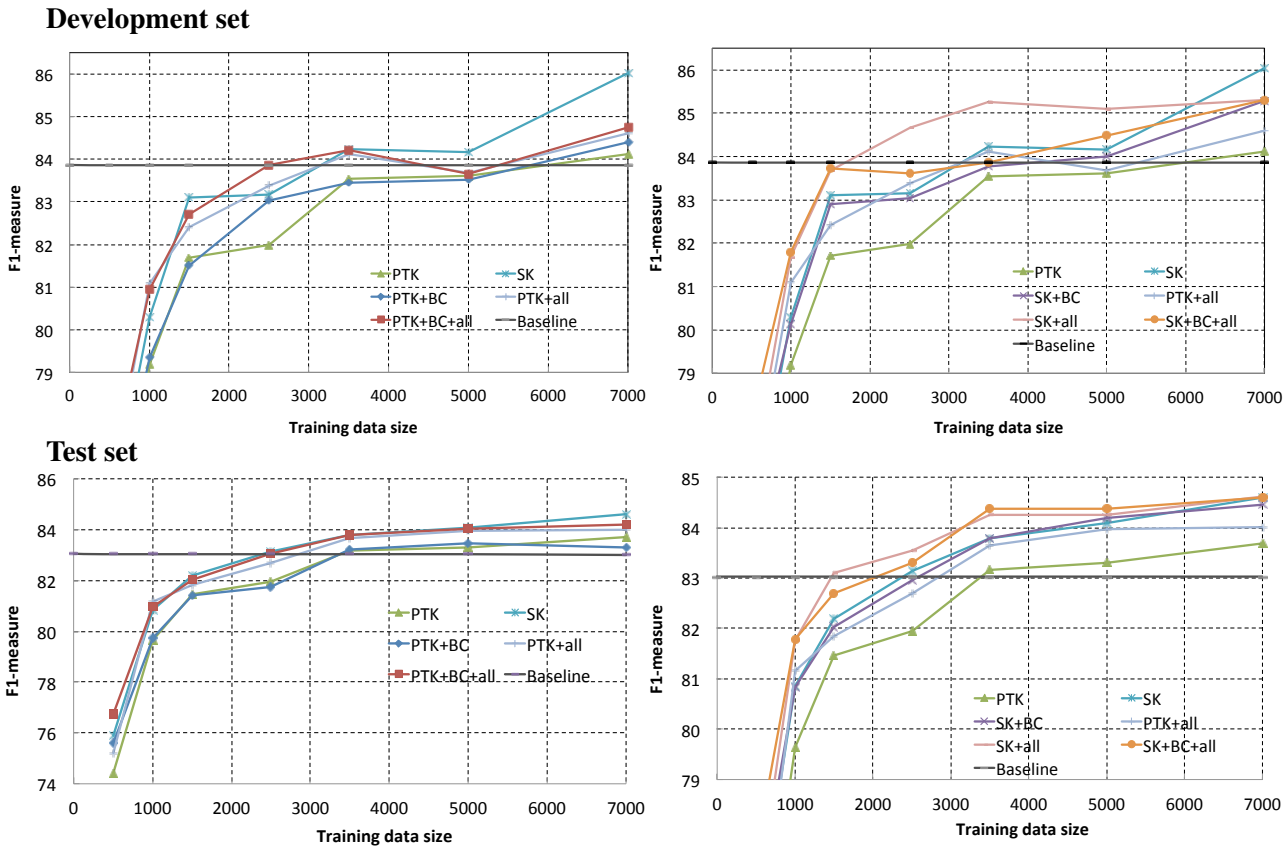


Figure 2: Learning curves for different reranking models on the development and on the testing sets.

The first two graphs in Fig. 2 show the plots on the development set whereas the last two are computed on the test set. The reranking models reported are Baseline, PTK, PTK+BC, PTK+all (features), PTK+BC+all, SK, SK+BC, SK+all and SK+BC+all.[8] We can see that: (*i*) PTK alone, i.e., without semantic information, has the lowest accuracy; (*ii*) BCs do not improve significantly any model; (*iii*) SK almost always achieves the highest accuracy; (*iv*) PTK+all (i.e., the model also using features) improves on PTK, but its accuracy is lower than for any model using SK, i.e., using semantic similarity; and (*v*) all features provide an initial boost to SK, but as soon as the data increases, their impact decreases.

5 Conclusion and Future Work

In summary, the learning curves clearly show the good generalization ability of SK, which improve the CRF baseline using little data (∼3,000). The semantic kernel significantly improves over the semi-CRF baseline and our previous state-of-the-art reranker exploiting shallow syntactic patterns (Saleh et al., 2014), which corresponds to PTK+all in the above comparison.

[8]Models are split between 2 plots in order to ease reading.

The improvement falls between 1-2 absolute percent points. This is remarkable as (*i*) it corresponds to ∼10% relative error reduction, and (*ii*) the state-of-the-art baseline system is very difficult to beat, as confirmed by the low impact of traditional features and BCs. Although the latter can generalize over concepts and words, their use is not straightforward, resulting in no improvement.

In the future, we plan to investigate the use of semantic similarity from distributional and other sources (Mihalcea et al., 2006; Padó and Lapata, 2007), e.g., Wikipedia (Strube and Ponzetto, 2006; Mihalcea and Csomai, 2007), Wiktionary (Zesch et al., 2008), WordNet (Pedersen et al., 2004; Agirre et al., 2009), FrameNet, VerbNet (Shi and Mihalcea, 2005), BabelNet (Navigli and Ponzetto, 2010), and LSA, and for different domains.

Acknowledgments

This research is part of the Interactive sYstems for Answer Search (Iyas) project, conducted by the Arabic Language Technologies (ALT) group at Qatar Computing Research Institute (QCRI) within the Qatar Foundation. We would like to thank Danilo Croce, Roberto Basili and Giuseppe Castellucci for helping and providing us with the similarity matrix for the semantic kernels.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado, June.

Roberto Basili, Marco Cammisa, and Alessandro Moschitti. 2006. A semantic kernel to classify texts with very few training examples. *Informatica (Slovenia)*, 30(2):163–172.

Stephan Bloehdorn and Alessandro Moschitti. 2007a. Combined syntactic and semantic kernels for text classification. In *Advances in Information Retrieval - Proceedings of the 29th European Conference on Information Retrieval (ECIR 2007)*, pages 307–318, Rome, Italy.

Stephan Bloehdorn and Alessandro Moschitti. 2007b. Structure and semantics for expressive text kernels. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM 2007)*, pages 861–864, Lisbon, Portugal.

Stephan Bloehdorn, Roberto Basili, Marco Cammisa, and Alessandro Moschitti. 2006. Semantic kernels for text classification based on topological measures of feature similarity. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 06)*, pages 808–812, Hong Kong.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Giuseppe Castellucci, Simone Filice, Danilo Croce, and Roberto Basili. 2013. UNITOR: Combining Syntactic and Semantic Kernels for Twitter Sentiment Analysis. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 369–374, Atlanta, Georgia, USA.

Nello Cristianini, John Shawe-Taylor, and Huma Lodhi. 2002. Latent Semantic Kernels. *Journal of Intelligent Information Systems*, 18(2):127–152.

Danilo Croce and Daniele Previtali. 2010. Manifold learning for the semi-supervised induction of framenet predicates: An empirical investigation. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, pages 7–16, Uppsala, Sweden.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011. Structured lexical similarity via convolution kernels on dependency trees. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1034–1046, Edinburgh, Scotland, UK.

Renato De Mori, Frederic Béchet, Dilek Hakkani-Tür, Michael McTear, Giuseppe Riccardi, and Gokhan Tur. 2008. Spoken Language Understanding. *IEEE Signal Processing Magazine*, 25:50–58.

Marco Dinarelli, Alessandro Moschitti, and Giuseppe Riccardi. 2012. Discriminative reranking for spoken language understanding. *IEEE Transactions on Audio, Speech and Language Processing*, 20(2):526–539.

G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, and K. E. Lochbaum. 1988. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '88)*, pages 465–480, New York, USA.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schlkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA, USA.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 282–289, Williamstown, MA, USA.

Ian McGraw, Scott Cyphers, Panupong Pasupat, Jingjing Liu, and Jim Glass. 2012. Automating crowd-supervised learning for spoken language systems. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association (INTERSPEECH 2012)*, pages 2473–2476, Portland, OR, USA.

Rada Mihalcea and Andras Csomai. 2007. Wikify! linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (CIKM 2007)*, pages 233–242, Lisbon, Portugal.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1 (AAAI 2006)*, pages 775–780, Boston, MA, USA.

Scott Miller, Richard Schwartz, Robert Bobrow, and Robert Ingria. 1994. Statistical Language Processing using Hidden Understanding Models. In *Proceedings of the workshop on Human Language Technology (HLT 1994)*, pages 278–282, Plainsboro, NJ, USA.

Alessandro Moschitti, Daniele Pighin, and Roberto Basili. 2006. Semantic role labeling via tree kernel

joint inference. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 61–68, New York City, USA.

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the 17th European Conference on Machine Learning (ECML 2006)*, pages 318–329, Berlin, Germany.

Alessandro Moschitti. 2010. Kernel engineering for fast and easy design of natural language applications. In *Coling 2010: Kernel Engineering for Fast and Easy Design of Natural Language Applications–Tutorial notes*, pages 1–91, Beijing, China.

Alessandro Moschitti. 2012. State-of-the-art kernels for natural language processing. In *Tutorial Abstracts of ACL 2012*, page 2, Jeju Island, Korea.

Alessandro Moschitti. 2013. Kernel-based learning to rank with syntactic and semantic structures. In *Tutorial abstracts of the 36th Annual ACM SIGIR Conference*, page 1128, Dublin, Ireland.

Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics (ACL 2010)*, pages 216–225, Uppsala, Sweden.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity - measuring the relatedness of concepts. In *HLT-NAACL 2004: Demonstration Papers*, pages 38–41, Boston, Massachusetts, USA.

Roberto Pieraccini, Esther Levin, and Chin-Hui Lee. 1991. Stochastic Representation of Conceptual Structure in the ATIS Task. In *Proceedings of the Fourth Joint DARPA Speech and Natural Language Workshop*, pages 121–124, Los Altos, CA, USA.

Christian Raymond and Giuseppe Riccardi. 2007. Generative and Discriminative Algorithms for Spoken Language Understanding. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH 2007)*, pages 1605–1608, Antwerp, Belgium, August.

Y. Dan Rubinstein and Trevor Hastie. 1997. Discriminative vs Informative Learning. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-1997)*, pages 49–53, Newport Beach, CA, USA.

Iman Saleh, Scott Cyphers, Jim Glass, Shafiq Joty, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2014. A study of using syntactic and semantic structures for concept segmentation and labeling. In *Proceedings of the 25th International Conference on Computational Linguistics*, COLING '14, pages 193–202, Dublin, Ireland.

G. Santafé, J.A. Lozano, and P. Larrañaga. 2007. Discriminative vs. Generative Learning of Bayesian Network Classifiers. In *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2007)*, pages 453–546, Hammamet, Tunisia.

Sunita Sarawagi and William W. Cohen. 2004. Semimarkov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems 17 (NIPS 2004)*, Vancouver, British Columbia, Canada.

Stephanie Seneff. 1989. TINA: A Probabilistic Syntactic Parser for Speech Understanding Systems. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-89)*, pages 711–714, Glasgow, UK.

Lei Shi and Rada Mihalcea. 2005. Putting pieces together: Combining framenet, verbnet and wordnet for robust semantic parsing. In *Computational Linguistics and Intelligent Text Processing*, pages 100–111. Springer Berlin Heidelberg.

Michael Strube and Simone Paolo Ponzetto. 2006. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06)*, pages 1419–1424, Boston, Massachusetts, USA.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)*, pages 173–180, Edmonton, Canada.

Cornelis J. van Rijsbergen. 1979. *Information Retrieval*. Butterworth-Heinemann Newton, MA, USA.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Using wiktionary for computing semantic relatedness. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI'08)*, pages 861–866, Chicago, Illinois,USA.