USING LEXICAL, SYNTACTIC AND SEMANTIC FEATURES FOR NON-TERMINAL GRAMMAR RULE INDUCTION IN SPOKEN DIALOGUE SYSTEMS

Georgia Athanasopoulou[1]*, Ioannis Klasinas*[1]*, Spiros Georgiladakis*[1]*, Elias Iosif*[2]*, Alexandros Potamianos*[2,3]

[1] School of ECE, Technical University of Crete, Chania 73100, Greece
[2] Athena Research and Innovation Center, Maroussi 15125, Greece
[3] School of ECE, National Technical University of Athens, Zografou 15780, Greece

gathanasopoulou@isc.tuc.gr, iklasinas@isc.tuc.gr, spgeo@intelligence.tuc.gr,
iosif.elias@gmail.com, potam@central.ntua.gr

ABSTRACT

In this work, we propose an algorithm for the automatic induction of non-terminal grammar rules for Spoken Dialogue Systems (SDS). Initially, a grammar developer provides the system with a minimal set of rules that serve as seeding examples. Using these seed rules and (optionally) a seed corpus, in-domain data are harvested and filtered from the web. A challenging task is identifying relevant chunks (phrases) in the web-harvested corpus that are good candidates for enhancing the seed grammar. We propose and evaluate rule-based and statistical classification algorithms for this purpose that use lexical, syntactic and semantic features. Induced grammars are evaluated in terms of accuracy of the proposed rules for two spoken dialogue domains. Results show up to four times absolute precision improvement compared to the naive grammar induction approach using semantic phrase similarity.

Index Terms— spoken language understanding,grammar induction, spoken dialogue systems, grammar enhancement

1. INTRODUCTION

The natural language understanding module of Spoken Dialogue Systems (SDS) models the underlying domain grammars. The manual construction of such grammars is a costly process, which requires human expertise. Tools for automatic (or semi-automatic) grammar induction facilitate the rapid development of SDS. Machine-assisted grammar induction has been an open research issue during the last two decades [1, 2, 3]; grammar induction algorithms can be broadly divided into two categories: supervised and unsupervised. Supervised algorithms rely on annotated data used for extracting features and training grammar rules or statistical parsers. Different types of training features are exploited that contain both syntactic [4, 5] and semantic [6, 7] information. Statistical parsers build probabilistic models for each set of grammar rules and select the most probable parse of a sentence [8]. They have been applied to various tasks such as the discovery of syntactic structure or the prediction of strings of words [9, 10, 11]. Often, supervised approaches suffer from limited portability due to the lack of training data for new domains and languages.

Unsupervised approaches alleviate the demand of annotated data, exploiting both syntactic [12, 13] and semantic features [14, 15]. These features are often extracted using language-specific tools such as shallow parsers or part-of-speech taggers. For example, in [16] an unsupervised approach is proposed for the automatic induction and filling of semantic slots that is adaptable to different domains and is built based on a (language specific) trained statistical semantic parser. This raises a barrier for under-resourced languages for which such tools are not available. Few unsupervised approaches have been proposed in the literature that are purely corpus-based and explore only lexical features extracted from a raw corpus, e.g., rule-based [17] or statistical-based [18, 19]. An effort towards open domain and fully unsupervised semantic parsing is presented in [20], where the required representations are constructed on-the-fly using distributional semantics.

In this work, we investigate a lightly supervised human-in-the-loop approach for corpus-based grammar induction. A developer provides a minimal set of examples (typically two to three relevant lexicalizations) for a grammar rule and then the system automatically suggests a set of fragments for enhancing each grammar rule. Our focus here is on non-terminal rules that are placed higher in the domain ontology and typically span two to five words. At the core of our work is an algorithm for the selection of lexical fragments (n-gram chunks) from a corpus that convey relevant semantic information in an unambiguous and concise manner. For example, consider the fragments "I want to depart from <city> on" and "depart from <city>" for the air travel domain. Both express the meaning of departure city, however, the semantics of the latter fragment are more concise and generalize better. Statistical and rule-based classification systems are proposed for the fragment selection problem. Each fragment selection system is then combined with a phrase-level semantic similar-

ity metric in order to induce a new set of grammar rules. The proposed systems are evaluated for the air travel and finance domains in two scenarios: training and testing on the same domain, as well as training on one domain and testing on the other. Although the proposed two-step grammar induction approach is evaluated on finite state machine grammars it could be also to statistical grammars/parsing.

2. SYSTEM ARCHITECTURE

An overview of the system architecture is depicted in Fig.1.

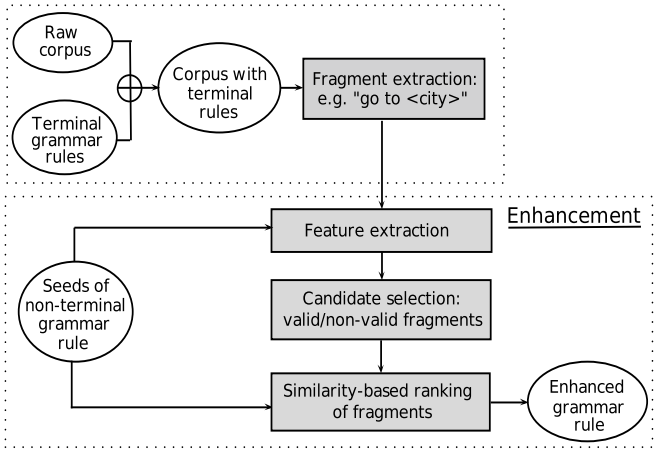

Fig. 1. *System architecture.*

The system uses a web-extracted corpus using the methodology described in [21] and comprises of the following components:

Induction of terminal rules: A terminal rule (or concept) resides in the leafs of the domain ontology, e.g., for the travel domain the concept <city> = ("New York", "Athens", "Salt Lake City", ...). The basic idea for the unsupervised induction of terminal rules relies on the distributional hypothesis of meaning, i.e., "similarity of context implies similarity of meaning" [22]. In this work, we hypothesize that such rules are already available using algorithms proposed in [21, 14].

Fragment extraction: This component extracts all phrase fragments from a corpus. Specifically, all n-grams built upon the domain concepts are extracted, with n ranging typically between two and five. For example, bigrams and trigrams for the sentence "travel from <city> tomorrow" include: {'from <city>', '<city> tomorrow', 'travel from <city>', 'from <city> tomorrow'}.

Grammar enhancement: The induction of non-terminal grammar rules constitutes the core of our work and is described through the *Enhancement* box in Fig.1. For each rule, seed examples are used for: 1) selecting a list with appropriate candidate fragments from the corpus, and 2) enhancing the rule using those candidates. An illustrative example is shown below, let the grammar rule <fromcity> and its corresponding seeding fragments be the following:

<fromcity>= ('leave from <city>', 'fly from <city>')

$\downarrow$ *After Enhancement*

<fromcity>= ('leave from <city>', 'fly from <city>', 'travel from <city>', 'flight from <airportname>', 'from <city>' 'depart from <city>', 'leaves from <state>', 'from <country>', 'take off from <city>', 'routes from <city>').

The grammar developer (fully or partially) confirms or discards the returned fragments and the enhanced grammar rule is formulated. The enhancement procedure can be repeated iteratively for every rule.

3. GRAMMAR ENHANCEMENT ALGORITHM

Let L correspond to the set of fragments exported from the *fragment extraction* step, r to one grammar rule and $\mathcal{F}_r = \{f_{r1}, ..., f_{r|\mathcal{F}_r|}\}$ to the set of seeding fragments of rule r provided by the developer (typically $|\mathcal{F}_r| = 2$ or 3). The seeding grammar rule r is enhanced in two steps: 1) selection of candidate fragments from list L by throwing away "junk" fragments that are poor candidates for the enhancement of r, and 2) ranking the list with candidates using a similarity metric in order to select the top e most appropriate candidates for enhancing r. The two steps can also be combined in a single ranking criterion, as discussed next.

For each fragment $f_i \in L,\ i = 1, .., |L|$ and a rule r we compute two scores: 1) the similarity score between r and fragment f_i, $S(r, f_i)$, and 2) the posterior probability that fragment f_i is a good candidate for enhancing grammar rule r, $P(r|f_i, \mathcal{F}_r)$. Given these two scores the enhancement steps are implemented as follows:

1. *Candidates' selection* from L, that is performed by setting a threshold th on $P(r|f_i, \mathcal{F}_r)$, i.e., if $P(r|f_i, \mathcal{F}_r) \leq th$ then f_i is removed. M_r is the resulting list of candidate fragments for rule r.

2. *Ranking* of the candidate fragments list, M_r, using the score $R(r, f_j)$ defined as the linear fusion of probability from the previous step and of the semantic similarity score $S(r, f_j)$:

$$R(r, f_j) = k \cdot P(r|f_i, \mathcal{F}_r) + (1 - k) \cdot S(r, f_i) \quad (1)$$

where $j = 1, ..., |M_r|$ with $|M_r| \leq |L|$ and $0 \leq k \leq 1$ is a factor that weights the influence of probability and similarity scores. The similarity score is computed as the average similarity between f_j and each seed $\mathcal{F}_r$ using the Longest Common Substring similarity metric [23]. The top e fragments are presented to the grammar developer.

In order to estimate the probability $P(r|f_i, \mathcal{F}_r)$, training data are required (i.e., full hand-crafted grammar rules). When no such data are available, a rule-based algorithm can be used for the candidates' selection step (detailed in Section 3.2) and then Eq. (1) can be applied with $k = 0$. Next, we present the features used in the statistical and rule-based approaches for the candidates' selection step.

3.1. Feature extraction

In this section, we present the features used for candidates' selection step. They can be divided in three main groups: 1) features extracted from corpus statistics (lexical features), 2) features relevant to the terminal concepts of each fragment [1] (syntactic and semantic features), 3) features estimated from the rule-specific seeding fragments $\mathcal{F}_r$. Specifically:
Features extracted from corpus statistics:

Let V be the vocabulary of the corpus and $c(\cdot)$ the function whose argument is an ordered set of words in V, and its value corresponds to the frequency of this set of words in corpus. Also, let $f = w_1\ w_2 .... \ w_t$ represent a fragment in L. The probability of fragment f is computed using statistical n-gram models [24] trained on the same in-domain corpus used for grammar induction. The perplexity of fragment f was used as feature, computed as:

$$perplexity(f) = 2^{H(f)}, \quad H(f) = \frac{1}{t} \cdot log_2(P(f)) \qquad (2)$$

In our experiments we used both features $log_2(P(f))$ and $perplexity(f)$ computed for $n = 2..4$, i.e., for bigram up to fourgram language models. An additional feature used was the probability $P_c(f)$ of fragment f computed as the number of occurrences of f normalized by the total number of occurrences of all fragments.

$$P_c(f) = \frac{c(f)}{\sum_{i=1}^{|L|} c(f_i)} \qquad (3)$$

Features extracted from terminal concepts in parsed corpus:

1. The ratio of terminal concepts over the total number of words in a fragment. For example, for the fragment 'traveling from <city>' the feature value is $\frac{1}{3}$.

2. To capture the relative position of terminal concepts in a fragment we computed the number of words following the last terminal in a fragment. For example, for the fragment 'traveling from <city> to', the feature value is 1.

Features extracted from seeding fragments:

The similarity between two fragments $f_q,\ f_r$ is estimated using each of two metrics: 1) $S_1(f_q, f_r)$: The longest common substring lexical similarity metric [23], and 2) $S_2(f_q, f_r)$: let l_a be the (character) length of the larger fragment (between f_q, f_r), l_b the length of the smaller fragment, $d = l_a - l_b$ the difference of the lengths and let $lev(f_q, f_r)$ be the function that computes the Levenshtein (or edit) distance of $f_q,\ f_r$ [25, 26], then the similarity of $f_q,\ f_r$ is computed as:

$$S_2(f_q, f_r) = \frac{l_a - lev(f_q, f_r)}{l_a + d} \qquad (4)$$

where the normalization of similarity score with the difference d, results in bigger similarity scores between fragments with approximately same length than with the ones with different lengths. To estimate the similarity between fragment f and the set of seeding rules $\mathcal{F}_r$ the average similarity between f and each of the seeding rules in $\mathcal{F}_r$ was computed and normalized by the average score of all fragments in L. Both $S_1(.,.)$ and $S_2(.,.)$ were used to estimate the average similarity features.

Other distances used to compare $f \in L$ with seeding fragments in $\mathcal{F}_r$ are: 1) if two fragments differ by a single word, the similarity score of different words, else 0, 2) indicator function that takes value 1 when f is a substring of a seeding fragments in $\mathcal{F}_r$, 3) f and a seed ends with the same terminal concept, 4) f and a seed have exactly the same lexical parts, 5) f is a substring of a seed with exactly one less word, 6) a seed is a substring of f with exactly one less word, e.g., f='depart from <city>' and f_{r1}='depart <city>', 7) statistics regarding the relative length of fragments, the number of concepts and of lexical parts.

3.2. Rule based selection of candidate fragments

This is a heuristic approach inspired by the manual process of grammar development and fragment selection. Based on how grammar developers perceive the validity of a fragment, we hand-crafted the decision tree shown in Fig. 2. The tree mainly employs features comparing each fragment, $f \in L$, with seeding fragments in $\mathcal{F}_r$. For each f, we start by checking the rules from the root of the tree until we reach a leaf. If the leaf corresponds to 1, the fragment is assigned a posterior $P(r|f, \mathcal{F}_r) = 1$, else it is pruned and assigned $P(r|f, \mathcal{F}_r) = 0$. This results in deterministically selecting candidates from list $|L|$ using only the rules shown in Fig. 2, i.e., independent from the threshold value, th. Thus, when using Eq. (1), only the similarity scores will influence the ranking among the selected candidate fragments, since $P(r|f, \mathcal{F}_r)$ will be equal to 1 for all of them.

3.3. Statistical selection of candidate fragments

Provided that we have an in-domain corpus and a corresponding hand-crafted grammar, it is easy to generate labeled training and test data for the candidates' selection step. Specifically, for the enhancement of a rule r only the candidate fragments that belong in r (in ground truth grammar) are labeled

[1] Terminal concepts are provided in the parsed corpus

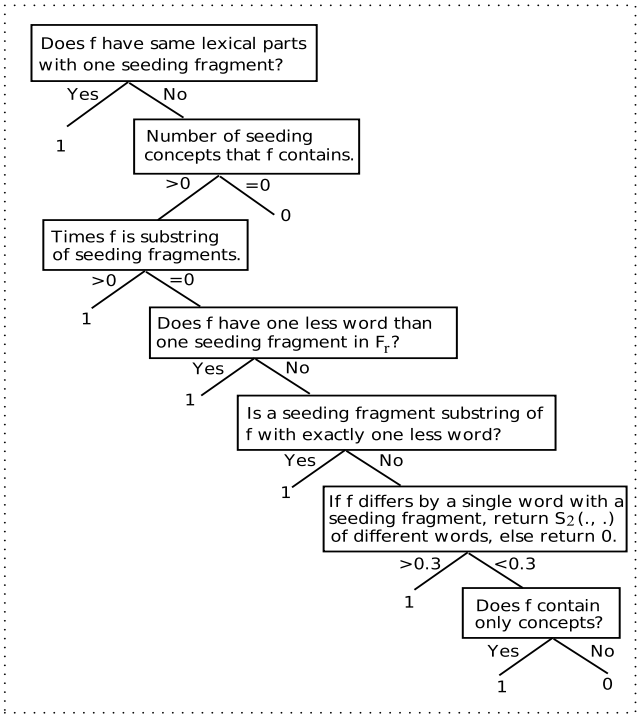

Fig. 2. *Rule based classification*

as 1, while all other fragments are labeled as 0. It is then easy to use machine learning to train a classifier using the features proposed in Section 3.1 on the training data (e.g. half the grammar rules) and test the performance of the grammar enhancement algorithm on the test set (the other half of the grammar rules).

Note that although the feature extraction process is dependent both on the in-domain corpus (for estimating language model probabilities and perplexity) and on the seed rules (for estimating similarities) the classifier is both domain and grammar rule independent. In fact, feature vectors extracted over all rules and fragments are used to train the classifier that performs the selection of candidate fragments. It is thus feasible (as shown next) to train a classifier for candidate fragments selection on one domain and test it on another with good results. The feature extraction process remains of course both domain and rule dependent.

For the statistical classification, a random forest classifier was used that has produced state-of-the-art results for similar tasks [27]. Boosting and SVM classifiers were also evaluated but not reported here due to lack of space.

4. EXPERIMENTAL PROCEDURE

The proposed system was evaluated with respect to two spoken dialogue system application domains: air travel and finance (currency exchange application). The main resources used for each domain were: a hand-crafted finite-state machine grammar and a web-harvested corpus. All resources are in English. The resources and their description can be downloaded from [28]. The process for web data harvesting and filtering is described in [21]. Note that for this work we assume that the terminal grammar rules have already been induced and hand-corrected by the grammar developer. The corpus is parsed using the terminal rules before fragment and feature extraction. The parsed corpus of the air travel domain consists of 56,765 utterances and for the finance domain of 14,935 utterances. A subset of the grammar rules that appear in each corpus were used for training and evaluation, specifically: 23 non-terminal concepts for the air-travel domain consisting of a total of 310 grammar rule fragments and 3 non-terminal concepts for the finance domain consisting of a total of 245 grammar rule fragments. The computation of frequencies, probabilities and perplexities for each domain was computed using the SRILM toolkit [29].

The data was organized for training and testing as follows: For the air travel domain, half of the concepts were selected for training the fragment candidates' selection algorithm, while the other half were used for evaluation. The criterion used for splitting 'equally' the rules into training and testing was the number of fragments of each rule. For the finance domain, the whole grammar was used for evaluation only. Thus, reported results are for matched domain training-testing conditions for travel and for mismatched for the finance domain (training on travel, testing on finance).

For each non-terminal rule, three randomly selected fragments were used as seeds, i.e., $|F_r| = 3$. The total number of (additional) requested fragments e per rule was 12. However, for rules r that had fewer fragments, e was set to the number of ground truth fragments of r (excluding the seeding fragments). The evaluation criterion used was the percentage of rule fragments induced that exactly matched[2] the fragments in the ground truth, i.e., fragment rule precision. The seed fragments were excluded from this calculation. Recall is not reported simply because a fixed number of fragments is requested for all experiments. The baseline system considers all fragments valid (a total of 74,590 valid bigrams to 5-grams for the travel domain and 91,061 for the finance domain) and simply uses semantic similarity to rank them. For the random forest classifier, we used 20 trees each randomly selecting 5 features. Results are averaged over five runs per rule to account for the random initialization of seed rule fragments, both training and testing.

5. RESULTS

The precision of the induced rules for the baseline and proposed system is shown in Fig. 3 and Fig. 4 for the travel and finance domain respectively. The performance is plotted as

[2]Note that a significant portion of the returned fragments were also relevant and could be edited into valid rule fragments, however, due to lack of space we do not report "partial" or "soft matches".

a function of the two parameters of the proposed algorithm, namely: the candidate fragments selection threshold th and the fusion weight k in Eq. (1). Specifically, we show results for three values (0, 0.15, and 0.3) of the candidates selection threshold th. In addition, the performance of the rule-based system is also displayed.

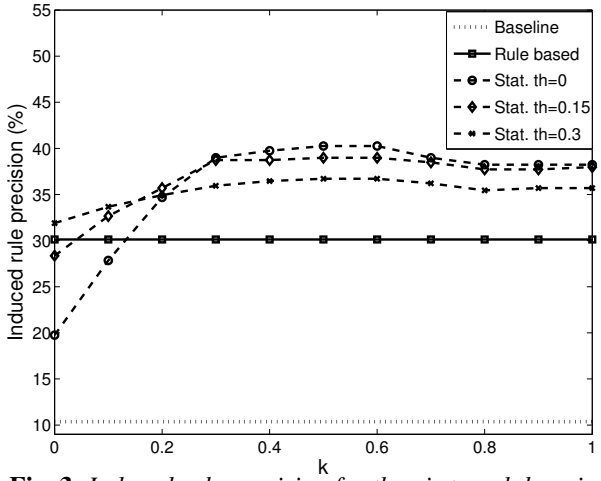

Fig. 3. *Induced rule precision for the air travel domain.*

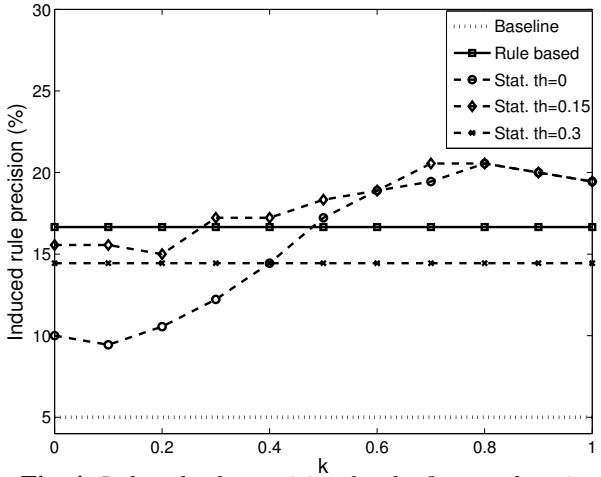

Fig. 4. *Induced rule precision for the finance domain.*

Introducing fragment candidate selection in the grammar induction algorithm significantly improves results for both the travel and finance domains. Specifically, rule-based *candidates' selection* outperforms the baseline by a factor of three boosting induction rule precision from around 10% to 30% for the travel domain, and from around 5% to 15% for the finance domain. Employing a statistical *candidate selection* algorithm further improves performance, reaching up to 41% precision for the travel domain and up to 22% precision for finance. Although the absolute performance is lower for the finance domain, the relative performance of each algorithm is consistent for both the travel and finance domains. The performance of the statistical *candidates' selection* algorithm is consistently high for a wide range of values of k (fusion parameter), giving best results in the range $k = 0.5$-0.7 for travel and $k = 0.7$-0.8 for finance. Best results are achieved around threshold parameter $th = 0.15$ for candidate fragments selection for both domains; this corresponds to pruning from L approximately 99.95% of fragments. For large threshold values, too few fragments survive the candidates' selection process leading to poor performance, e.g., using threshold $th = 0.3$ for the finance domain. In general, the parameters k and th can be optimized on held-out data, but further experimentation is needed to investigate the generalizability across different domains. Overall, the proposed grammar induction algorithm significantly improves over the baseline for both evaluation scenarios, i.e., training and testing on travel, training on travel and testing on finance.

6. CONCLUSIONS

We proposed a lightly supervised corpus-based algorithm for the automatic induction of non-terminal grammar rules for spoken dialogue systems. The algorithm was bootstrapped by a minimal set of seed examples for each grammar rule. Grammar induction was realized as a two-step process: *candidates' selection* from the list of all corpus fragments using a set of lexical, syntactic and semantic features, followed by *ranking* the candidates via a semantic similarity metric.

The proposed algorithm significantly outperforms a naive grammar induction approach that uses only a semantic similarity metric to rank candidates. For both the air travel and finance domains, enhanced fragments' accuracy is increased by a factor of four using the proposed approach. Furthermore, the statistical fragment candidate selection algorithm significantly outperforms the rule-based one. Good performance is obtained even with a limited number of seeding examples, e.g., three. Most importantly, the algorithm is shown to be portable across application domains, although absolute performance is lower when training on one domain and testing on another. Portability is important for the rapid development of grammars in new domains where limited linguistic resources are available.

In future work we will investigate additional features for improved classification performance, as well as experiment with more domain and languages to verify the portability of the proposed algorithms. In addition, we will investigate the relevance of the proposed approach to statistical parsing applications, where limited training data is available.

7. ACKNOWLEDGEMENTS

This work was partially funded by the PortDial (grant number 296170) and SpeDial (grant number 611396) projects supported by the EU Seventh Framework Programme (FP7). The authors wish to thank Maria Vomva and Maria Giannoudaki for the development of ground truth grammars.

8. REFERENCES

[1] E. Brill, "A simple rule-based part of speech tagger," in *Proceedings of the Workshop on Speech & Natural Language*, 1992.

[2] K. Lari and S.J. Young, "The estimation of stochastic context-free grammars using the inside-outside algorithm," *Computer Speech & Language*, vol. 4, no. 1, pp. 35 – 56, 1990.

[3] S. F. Chen, "Bayesian grammar induction for language modeling," in *Proceedings of Annual meeting - Association for Computational Linguistics*, 1995.

[4] R. Hwa, "Supervised grammar induction using training data with limited constituent information," in *Proceedings of Annual meeting - Association for Computational Linguistics*, 1999.

[5] R. Hwa, "Sample selection for statistical grammar induction," in *Proceedings of the Joint SIGDAT*, 2000.

[6] R. J. Kate and R. J. Mooney, "Semi-supervised learning for semantic parsing using support vector machines," in *Proceedings of Human Language Technologies*, 2007.

[7] S. S. Pradhan, W. Ward, K. Hacioglu, J. H. Martin, and D. Jurafsky, "Shallow semantic parsing using support vector machines.," in *Proceedings of HLT-NAACL*, 2004.

[8] M. Collins, "Head-driven statistical models for natural language parsing," *Computational linguistics*, vol. 29, no. 4, pp. 589–637, 2003.

[9] E. Charniak, "A maximum-entropy-inspired parser," in *Proceedings of* 1^{st} *North American chapter - Association for Computational Linguistics*, 2000.

[10] M. Lease, E. Charniak, and M. Johnson, "Parsing and its applications for conversational speech," in *Proceedings of ICASSP*, 2005.

[11] B. Roark, "Probabilistic top-down parsing and language modeling," *Computational linguistics*, vol. 27, no. 2, pp. 249–276, 2001.

[12] D. Klein and C. D. Manning, "Corpus-based induction of syntactic structure: models of dependency and constituency," in *Proceedings of Annual meeting - Association for Computational Linguistics*, 2004.

[13] S. B. Cohen and N. A. Smith, "Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction," in *Proceedings of Human Language Technologies*, 2009.

[14] H. H. Meng and K. C. Siu, "Semiautomatic acquisition of semantic structures for understanding domain-specific natural language queries," *IEEE Trans. on Knowl. & Data Eng.*, vol. 14, no. 1, pp. 172–181, 2002.

[15] H. Poon and P. Domingos, "Unsupervised semantic parsing," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2009.

[16] Y. N. Chen, W. Y. Wang, and A. I. Rudnicky, "Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing," in *Proceedings of ASRU, IEEE Workshop*, 2013.

[17] Y. Seginer, "Fast unsupervised incremental parsing," in *Proceedings of Annual Meeting-Association for Computational Linguistics*, 2007.

[18] E. Ponvert, J. Baldridge, and K. Erk, "Simple unsupervised grammar induction from raw text with cascaded finite state models.," in *Proceedings of ACL*, 2011.

[19] Y. Bisk and J. Hockenmaier, "Simple robust grammar induction with combinatory categorial grammars.," in *Proceedings of AAAI*, 2012.

[20] I. Beltagy, K. Erk, and R. Mooney, "Semantic parsing using distributional semantics and probabilistic logic," in *Proceedings of ACL Workshop*, 2014b.

[21] I. Klasinas, A. Potamianos, E. Iosif, S. Georgiladakis, and G. Mameli, "Web data harvesting for speech understanding grammar induction," in *Proceedings of Interspeech*, 2013.

[22] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 23, pp. 146–162, 1954.

[23] G. Stoilos, G. Stamou, and S. Kollias, "A string metric for ontology alignment," in *The Semantic Web–ISWC*, pp. 624–637. Springer, 2005.

[24] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, Prentice-Hall, 2000.

[25] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," in *Proceedings of Sov. Phys. Dokl.*, 1966.

[26] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," *Journal of the ACM (JACM)*, vol. 21, no. 1, pp. 168–173, 1974.

[27] K. J. Archer and R. V. Kimes, "Empirical characterization of random forest variable importance measures," *Computational Statistics & Data Analysis*, vol. 52, no. 4, pp. 2249–2260, 2008.

[28] "PortDial project, free data deliverable D3.1," https://sites.google.com/site/portdial2/deliverables-publications, [Online].

[29] A. Stolcke, "Srilm-an extensible language modeling toolkit.," in *Proceedings of Interspeech*, 2002.