SemEval-2014 Task 2: Grammar Induction for Spoken Dialogue Systems

Ioannis Klasinas[1], **Elias Iosif**[2,4], **Katerina Louka**[3], **Alexandros Potamianos**[2,4]
[1] School of ECE, Technical University of Crete, Chania 73100, Greece
[2] School of ECE, National Technical University of Athens, Zografou 15780, Greece
[3] Voiceweb S.A., Athens 15124, Greece
[4] Athena Research Center, Marousi 15125, Greece
iklasinas@isc.tuc.gr,{iosife,potam}@telecom.tuc.gr,klouka@voiceweb.eu

Abstract

In this paper we present the SemEval-2014 Task 2 on spoken dialogue grammar induction. The task is to classify a lexical fragment to the appropriate semantic category (grammar rule) in order to construct a grammar for spoken dialogue systems. We describe four subtasks covering two languages, English and Greek, and three speech application domains, travel reservation, tourism and finance. The classification results are compared against the groundtruth. Weighted and unweighted precision, recall and f-measure are reported. Three sites participated in the task with five systems, employing a variety of features and in some cases using external resources for training. The submissions manage to significantly beat the baseline, achieving a f-measure of 0.69 in comparison to 0.56 for the baseline, averaged across all subtasks.

1 Introduction

This task aims to foster the application of computational models of lexical semantics to the field of spoken dialogue systems (SDS) for the problem of grammar induction. Grammars constitute a vital component of SDS representing the semantics of the domain of interest and allowing the system to correctly respond to a user's utterance.

The task has been developed in tight collaboration between the research community and commercial SDS grammar developers, under the auspices of the EU-IST PortDial project[1]. Among the project aims is to help automate the grammar development and localization process. Unlike previous approaches (Wang and Acero, 2006; Cramer, 2007) that have focused on full automation, PortDial adopts a human-in-the-loop approach were a developer bootstraps each grammar rule or request type with a few examples (use cases) and then machine learning algorithms are used to propose grammar rule enhancements to the developer. The enhancements are post-edited by the developer and new grammar rule suggestions are proposed by the system, in an iterative fashion until a grammar of sufficient quality is achieved. In this task, we focus on a snapshot of this process, where a portion of the grammar is already induced and post-edited by the developer and new candidate fragments are rolling in order to be classified to an existing rule (or rejected). The goal is to develop machine learning algorithms for classifying candidate lexical fragments to the correct grammar rule (semantic category). The task is equally relevant for both finite-state machine and statistical grammar induction.

In this task the semantic hierarchy of SDS grammars has two layers, namely, low- and high-level. Low-level rules are similar to gazetteers referring to terminal concepts that can be as represented as sets of lexical entries. For example, the concept of city name can be represented as <CITY> = ("London", "Paris", ...). High-level rules are defined on top of low-level rules, while they can be lexicalized as textual fragments (or chunks), e.g., <TOCITY> = ("fly to <CITY>", ...). Using the above examples the sentence "I want to fly to Paris" will be first parsed as "I want to fly to <CITY>" and finally as "I want to <TOCITY>".

In this task, we focus exclusively on high-level rule induction, assuming that the low-level rules are known. The problem of fragment extraction and selection is simplified by investigating the

[1] http://www.portdial.eu/

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 9–16,
Dublin, Ireland, August 23-24, 2014.

binary classification of (already extracted) fragments into valid and non-valid. The task boils down mainly to a semantic similarity estimation problem for the assignment of valid fragments into high-level rules.

2 Prior Work

The manual development of grammars is a time-consuming and tedious process that requires human expertise, posing an obstacle to the rapid porting of SDS to new domains and languages. A semantically coherent workflow for SDS grammar development starts from the definition of low-level rules and proceeds to high-level ones. This process is also valid for the case of induction algorithms. Automatic or machine-aided grammar creation for spoken dialogue systems can be broadly divided in two categories (Wang and Acero, 2006): knowledge-based (or top-down) and data-driven (or bottom-up) approaches.

Knowledge-based approaches rely on the manual or semi-automatic development of domain-specific grammars. They start from the domain ontology (or taxonomy), often in the form of semantic frames. First, terminal concepts in the ontology (that correspond to low-level grammar rules) get populated with values, e.g., <CITY>, and then high-level concepts (that correspond to high-level grammar rules) get lexicalized creating grammar fragments. Finally, phrase headers and trailers are added to create full sentences. The resulting grammars often suffer from limited coverage (poor recall). In order to improve coverage, regular expressions and word/phrase order permutations are used, however at the cost of over-generalization (poor precision). Moreover, knowledge-based grammars are costly to create and maintain, as they require domain and engineering expertise, and they are not easily portable to new domains. This led to the development of grammar authoring tools that aim at facilitating the creation and adaptation of grammars. SGStudio (Semantic Grammar Studio), (Wang and Acero, 2006), for example, enables 1) example-based grammar learning, 2) grammar controls, i.e., building blocks and operators for building more complex grammar fragments (regular expressions, lists of concepts), and 3) configurable grammar structures, allowing for domain-adaptation and word-spotting grammars. The Grammatical Framework Resource Grammar Library (GFRGL) (Ranta, 2004) enables the creation of multilingual grammars adopting an abstraction formalism, which aims to hide the linguistic details (e.g., morphology) from the grammar developer.

Data-driven approaches rely solely on corpora (bottom-up) of transcribed utterances (Meng and Siu, 2002; Pargellis et al., 2004). The induction of low-level rules consists of two steps dealing with the 1) identification of terms, and 2) assignment of terms into rules. Standard tokenization techniques can be used for the first step, however, different approaches are required for the case of multiword terms, e.g., "New York". In such cases, gazetteer lookup and named entity recognition can be employed (if the respective resources and tools are available), as well as corpus-based collocation metrics (Frantzi and Ananiadou, 1997). Typically, the identified terms are assigned into low-level rules via clustering algorithms operating over a feature space that is built according to the term semantic similarity. The distributional hypothesis of meaning (Harris, 1954) is a widely-used approach for estimating term similarity. A comparative study of similarity metrics for the induction of SDS low-level rules is presented in (Pargellis et al., 2004), while the combination of metrics was investigated in (Iosif et al., 2006). Different clustering algorithms have been applied including hard- (Meng and Siu, 2002) and soft-decision (Iosif and Potamianos, 2007) agglomerative clustering.

High-level rule induction is a less researched area that consists of two main sub-problems: 1) the extraction and selection of candidate fragments from a corpus, and 2) assignment of terms into rules. Regarding the first sub-problem, 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. The application of syntactic parsers for segment extraction is not straightforward since the output is a full parse tree. Moreover, such parsers are typically trained over annotated corpora of formal language usage, while the SDS corpora often are ungrammatical due to spontaneous speech. There are few statistical parsing algorithms that rely only on plain lexical features (Ponvert et al., 2011; Bisk and Hockenmaier, 2012) however, as other algorithms, one needs to decide where to prune the parse tree. In (Georgiladakis et al., 2014), the explicit extraction and selection of fragments is investigated following an example-driven approach where few rule seeds are provided by the grammar developer. The second sub-problem of high-level rule induction deals with the formulation of rules using the selected fragments. Each rule is meant to consist of semantically similar fragments. For this purpose, clustering algorithms can be employed exploiting the semantic similarity between fragments as features. This is a challenging problem since the fragments are multi-word structures whose overall meaning is composed according to semantics of the individual constituents. Recently, several models have been proposed regarding phrase (Mitchell and Lapata, 2010) and sentence similarity (Agirre et al., 2012), while an approach towards addressing the issue of semantic compositionality is presented in (Milajevs and Purver, 2014).

The main drawback of data-driven approaches is the problem of data sparseness, which may affect the coverage of the grammar. A popular solution to the data sparseness bottleneck is to harvest in-domain data from the web. Recently, this has been an active research area both for SDS systems and language modeling in general. Data harvesting is performed in two steps: (i) query formulation, and (ii) selection of relevant documents or sentences (Klasinas et al., 2013). Posing the appropriate queries is important both for obtaining in-domain and linguistically diverse sentences. In (Sethy et al., 2007), an in-domain language model was used to identify the most appropriate n-grams to use as web queries. An in-domain language model was used in (Klasinas et al., 2013) for the selection of relevant sentences. A more sophisticated query formulation was proposed in (Sarikaya, 2008), where from each in-domain utterance a set of queries of varying length and complexity was generated. These approaches assume the availability of in-domain data (even if limited) for the successful formulation of queries; this dependency is also not eliminated when using a mildly lexicalized domain ontology to formulate the queries, as in (Misu and Kawahara, 2006). Selecting the most relevant sentences that get returned from web queries is typically done using statistical similarity metrics between in domain data and retrieved documents, for example the BLEU metric (Papineni et al., 2002) of n-gram similarity in (Sarikaya, 2008) and a metric of relative entropy (Kullback-Leibler) in (Sethy et al., 2007). In cases where in-domain data is not available, cf. (Misu and Kawahara, 2006), heuristics (pronouns, sentence length, wh-questions) and matches with out-of-domain language models can be used to identify sentences for training SDS grammars. In (Sarikaya, 2008), the produced grammar fragments are also parsed and attached to the domain ontology. Harvesting web data can produce high-quality grammars while requiring up to 10 times less in-domain data (Sarikaya, 2008).

Further, data-driven approaches induce syntactic grammars but do not learn their corresponding meanings, for this purpose an additional step is required of parsing the grammar fragments and attaching them to the domain ontology (Sarikaya, 2008). Also, in many cases it was observed that the fully automated bottom-up paradigm results to grammars of moderate quality (Wang and Acero, 2006), especially on corpora containing longer sentences and more lexical variety (Cramer, 2007). Finally, algorithms focusing on crosslingual grammar induction, like CLIoS (Kuhn, 2004), are often even more resource-intensive, as they require training corpora of parallel text and sometimes also a grammar for one of the languages. Grammar quality can be improved by introducing a human in the loop of grammar induction (Portdial, 2014a); an expert that validates the automatically created results (Meng and Siu, 2002).

3 Task Description

Next we describe in detail the candidate grammar fragment classification SemEval task. This task is part of a grammar rule induction scenario for high-level rules. The evaluation focuses in spoken dialogue system grammars for multiple domains and languages.

3.1 Task Design

The goal of the task is to classify a number fragment to the rules available in the grammar. For each grammar we provide a training and development set, i.e., a set of rules with the associated fragments and the test set which is composed of plain fragments. An excerpt of the train set for the rule “<TOCITY>” is “ARRIVE AT <CITY>, ARRIVES AT <CITY>, GOING TO <CITY>” and of the test set “GOING INTO <CITY>, AR-

RIVES INTO <CITY>".

In preliminary experiments during the task design we noticed that if the test set consists of valid fragments only, good classification performance is achieved, even when using the naive baseline system described later in this paper. To make the task more realistic we have included a set of *"junk"* fragments not corresponding to any specific rule. Junk fragments were added both in the train set where they are annotated as such and in the test set. For this task we have artificially created the junk fragments by removing or adding words from legitimate fragments. Example junk fragments used are "HOLD AT AT <TIME> TRY" and "ANY CHOICE EXCEPT <AIRLINE> OR", the first one having a repetition of the word "AT" while the second one should include one more time the concept "<AIRLINE>" in the end to be meaningful.

Junk fragments help better model a real-world scenario, where the candidate fragments will include irrelevant examples too. For example, if web corpora are used to extract the candidate fragments grammatical mistakes and out-of-domain sentences might appear. Similarly, if the transcriptions from a deployed SDS system are used for grammar induction, transcription errors might introduce noise (Bechet et al., 2014).

Junk fragments account for roughly 5% of the train test and 15% of the test set. The discrepancy between train and test set ratios is due to a conscious effort to model realistic train/test conditions, where train data is manually processed and does not include errors, while candidate fragments are typically more noisy.

3.2 Datasets

We have provided four datasets, travel English, travel Greek, tourism English and finance English. The travel domain grammar covers flight, car and hotel reservation utterances. The tourism domain covers touristic information including accommodation, restaurants and movies. The finance domain covers utterances of a bank client asking questions about his bank account as well as reporting problems. In Table 1 are presented typical examples of fragments for every subtask.

All grammars have been manually constructed by a grammar developer. For the three English grammars, a small corpus (between 500 and 2000 sentences) was initially available. The grammar developer first identified terminal concepts, which correspond to low-level rules. Typical examples include city names for the travel domain, restaurant names for the tourism domain and credit card names in the finance domain. After covering all low-level rules the grammar developer proceeded to identify high-level rules present in the corpus, like the departure city in the travel domain, or the user request type for a credit card. The grammar developer was instructed to identify all rules present in the corpus, but also spend some effort to include rules not appearing in the corpus so that the resulting grammar better covers the domain at hand. For the case of Greek travel grammar no corpus was initially available. The Greek grammar was instead produced by manually translating the English one, accounting for the differences in syntax between the two languages. The grammars have been developed as part of the PortDial FP7 project and are explained in detail in (Portdial, 2014b).

For the first three datasets that have been available from the beginning of the campaign we have split the release into train, development and test set. For the finance domain which was announced when the test sets were released we only provided the train and test set, to simulate a resource poor scenario. The statistics of the datasets for all language/domain pairs are given in Table 2.

In addition to the high-level rules we made available the low-level rules for each grammar, which although not used in the evaluation, can be useful for expanding the high-level rules to cover all lexicalizations expressed by the grammar.

3.3 Evaluation

For the evaluation of the task we have used precision, recall and f-measure, both weighted and unweighted.

If R_j denotes the set of fragments for one rule and C_j the set of fragments classified to this rule by a system then per-rule precision is computed by the equation:

$$Pr_j = \frac{|R_j \cap C_j|}{|C_j|}$$

and per-rule recall by:

$$Rc_j = \frac{|R_j \cap C_j|}{|R_j|}$$

F-measure is then computed by:

Grammar	Rule	Fragment
Travel English	<FLIGHTFROM>	FLIGHT FROM <CITY>
Travel Greek	<FLIGHTFROM>	ΠΤΗΣΗ ΑΠΟ <CITY>
Tourism English	<TRANSFERQ>	TRANSFERS FROM <airportname> TO <cityname>
Finance English	<CARDNAME>	<BANKNAME> CARD

Table 1: Example grammar fragments for each application domain.

Grammar	Rules	Fragments		
		Train set	Dev set	Test set
Travel English	32	623	331	284
Travel Greek	35	616	340	324
Tourism English	24	694	334	285
Finance English	9	136	-	37

Table 2: Number of rules in the training, development and test sets for each application domain.

$$F_j = \frac{2Pr_jRc_j}{Pr_j + Rc_j}$$

.

Precision for all the J rules $R_j, 1 \leq j \leq J$ is computed by the following equation:

$$Pr = \sum_j Pr_j w_j$$

In the unweighted case the weight w_j has a fixed value for all rules, so $w_j = \frac{1}{J}$. Taking into account the fact that the rules are not balanced in terms of fragments, a better way to compute for the weight is $w_j = \frac{|R_j|}{\sum_j |R_j|}$. In the latter, weighted, case the total precision will better describe the results.

Recall is similarly computed using the same weighting scheme as:

$$Rc = \sum_j Rc_j w_j$$

3.4 Baseline

For comparison purposes we have developed a naive baseline system. To classify a test fragment, first its similarity with all the train fragments is computed, and it is classified to the rule where the most similar train fragment belongs. Fragment similarity is computed as the ratio of their Longest Common Substring (LCS) divided by the sum of their lengths:

$$Sim(s,t) = \frac{|LCS(s,t)|}{|s| + |t|}$$

where s and t are two strings, $|s|$ and $|t|$ their length in characters and $|LCS(s,t)|$ the length of their LCS. This is a very simple baseline, computing similarity without taking into account context or semantics.

4 Participating Systems

Three teams have participated in the task with five systems. All teams participated in all subtasks with the exception of travel Greek, where only two teams participated. An overview of core system features is presented in Table 3. The remainder of this section briefly describes each of the submissions and then compares them. A brief description for each system is provided in the following paragraphs.

tucSage. The core of the tucSage system is a combination of two components. The first component is used for the selection of candidate rule fragments from a corpus. Specifically, the posterior probability of a candidate fragment belonging to a rule is computed using a variety of features. The feature set includes various lexical features (e.g., the number of tokens), the fragment perplexity computed using n-gram language modeling, and features based on lexical similarity. The second component is used for computing the similarity between a candidate fragment and a grammar rule. In total, two different types of similarity metrics are used relying on the overlap of character bigrams and contextual features. These similarities are fused with the posterior probabilities produced by the fragment selection model. The contribution of the two components is adjusted using an exponential weight.

SAIL-GRS. The SAIL-GRS system is based on the well-established term frequency–inverse document frequency ($TF-IDF$) measurement. This metric is adapted to the present task by considering each grammar rule as a “document”. For each rule, all its fragments are aggregated

System acronym	Use of machine learn.	Features used	Similarity metrics	External corpora	Language-specific
Baseline	no	lexical	Longest Common Substring	no	no
tucSage	yes: random forests	lexical, perplexity, similarity-based , heuristic	character overlap, cosine similarity	web documents	no
SAIL-GRS	no	lexical	cosine similarity	no	no
Biel	no	lexical, expansion of low-level rules	cosine similarity	Wikipedia articles	yes

Table 3: Overview of the characteristics of the participating systems.

and the frequency of the respective n-grams (constituents) is computed. The inverse document frequency is casted as inverse rule frequency and it is computed for the extracted n-grams. The process is performed for both unigrams and bigrams.

Biel. The fundamental idea behind the Biel system is the encoding of domain semantics via topic modeling. For this purpose a background document space is constructed using thousands of Wikipedia articles. Particular focus is given to the transformation of the initial document space according to the paradigm of explicit semantic analysis. For each domain, a topic space is defined and a language-specific function is employed for the mapping of documents. In essence, the mapping function is an association measurement that is based on $TF-IDF$ scores. An approximation regarding the construction of the topic space is investigated in order to reduce data sparsity, while a number of normalization schemes are also presented.

Overall, only the tucSage system employs a machine learning-based approach (random forests), while an unsupervised approach is followed by the SAIL-GRS and Biel systems. All systems exploit lexical information extracted from rule fragments. This information is realized as the lexical surface form of the constituents of fragments. For example, consider the "depart for <CITY>" fragment that corresponds to the high-level rule referring to the notion of departure city. The following set of lexical features can be extracted from the aforementioned fragment: ("depart", "from", "<CITY>"). Unlike the other systems, the Biel system utilizes low-level rules to expand high-level rules with terminal concept instances. For example, the "<CITY>" rule is not processed as is, but it is represented as a list of city names ("New York", "Boston", ...). The most rich feature set is used by the tucSage system which combines lexical, perplexity and similarity features with a set of heuristic rules. All three systems employ the widely-used cosine similarity metric. Both SAIL-GRS and Biel systems rely solely on this metric during the assignment of an unknown fragment to a high-level rule. A more sophisticated approach is presented by tucSage, where first a classifier is built for every grammar rule, computing the probability of a fragment belonging to this rule and then the similarity between the fragment and the rule is computed. Classification is then performed by combining the two scores. Also, another difference regarding the employment of the cosine similarity deals with the computation of the vectorial feature values. A simple binary scheme is used in the tucSage system, while variations of the term frequency-inverse document frequency scheme are used in SAIL-GRS and Biel. Besides cosine similarity, a similarity metric based on the overlap of character bigrams is used by the tucSage system. External corpora (i.e., corpora that were not provided as part of the official task data) were used by the tucSage and Biel systems. Such corpora were meant as an additional source of information with respect to the domains under investigation. Regarding tucSage, the training data were exploited in order to construct web search queries for harvesting a collection of web documents from which a number of sentences were selected for corpus creation. In the case of the Biel system, a set of Wikipedia articles was exploited. Language specific resources where used for the Biel system, while the other two teams used language agnostic methods.

5 Results

The results for all participating teams and the baseline system are given in Table 4. The tucSage team submitted three runs, the first one being the primary, indicated with an asterisk in the results.

Focusing on the weighted F-measure we see that in all domains but the tourism English, at least one submission manages to outperform the baseline provided by the organizers. In travel English the baseline system achieves 0.51 weighted f-measure, with two out of the three systems achieving 0.68 and 0.58. The improvement over the baseline is greater for the travel Greek subtask, where the baseline score of 0.26 is much lower than the achieved 0.52 from tucSage. In the tourism English subtask the best submitted systems managed to match the performance of the baseline system, but not to exceed it. This can be attributed to the good performance of the baseline system, due to the fact that the tourism grammar is composed of longer fragments than the rest, helping the naive baseline system achieve top performance exploiting lexical similarity only. We can however assume that more complex systems would beat the baseline if the test set fragments were built using different lexicalizations, as would be the case in unannotated data coming from deployed SDS.

In the finance domain, even though the amount of training data is quite smaller than in all other subtasks the submitted systems still manage to outperform the baseline system. This means that the submitted systems display robust performance both in resource-rich and resource-poor conditions.

6 Conclusion

The tucSage and SAIL-GRS systems are shown to be portable across domains and languages, achieving performance that exceeds the baseline for three out of four datasets. The highest performance of the tucSage system compared to the SAIL-GRS system may be attributed to the use of a model for fragment selection. Interestingly, the simple variation of the *TF−IDF* scheme used by the SAIL system achieved very good results being a close second performer. The UNIBI system proposed a very interesting new application of the framework of topic modeling to the task of grammar induction, however, the respective performance does not exceed the state-of-the-art. The combination of the tucSage and SAIL-GRS systems could give better results.

team	Weighted			Unweighted		
	Pr.	Rec.	F-m.	Pr.	Rec.	F-m.
			Travel English			
Baseline	0.40	0.69	0.51	0.38	0.67	0.48
tucSage1*	0.60	**0.73**	0.66	0.59	**0.74**	0.66
tucSage2	0.59	0.72	0.65	0.59	**0.74**	0.65
tucSage3	**0.69**	0.67	**0.68**	**0.66**	0.69	**0.67**
SAIL-GRS	0.54	0.62	0.58	0.57	0.66	0.61
Biel	0.13	0.39	0.20	0.09	0.34	0.14
			Travel Greek			
Baseline	0.17	**0.65**	0.26	0.16	**0.73**	0.26
tucSage1*	0.47	0.58	**0.52**	**0.55**	0.72	**0.62**
tucSage2	0.46	0.53	0.49	0.50	0.59	0.54
tucSage3	**0.51**	0.48	0.49	0.52	0.56	0.54
SAIL-GRS	0.46	0.51	0.49	0.49	0.62	0.55
Biel	-	-	-	-	-	-
			Tourism English			
Baseline	**0.80**	**0.94**	**0.87**	**0.82**	**0.94**	**0.87**
tucSage1*	0.79	**0.94**	0.86	0.76	0.91	0.83
tucSage2	0.78	0.93	0.85	0.73	0.90	0.80
tucSage3	**0.80**	0.93	0.86	0.77	0.90	0.83
SAIL-GRS	0.75	0.90	0.82	0.75	0.90	0.82
Biel	0.04	0.14	0.06	0.02	0.08	0.04
			Finance English			
Baseline	0.48	0.78	0.60	0.40	**0.63**	0.49
tucSage1*	0.61	**0.81**	0.70	0.43	0.54	0.48
tucSage2	0.55	0.74	0.63	0.40	0.51	0.45
tucSage3	0.52	0.67	0.58	0.39	0.43	0.41
SAIL-GRS	**0.78**	0.78	**0.78**	**0.67**	0.62	**0.65**
Biel	0.22	0.30	0.25	0.06	0.18	0.09
			Average over all four tasks			
Baseline	0.46	0.73	0.56	0.44	**0.74**	0.53
tucSage1*	0.62	**0.77**	**0.69**	0.58	0.73	0.65
tucSage2	0.60	0.73	0.66	0.56	0.69	0.61
tucSage3	**0.63**	0.69	0.65	0.59	0.65	0.61
SAIL-GRS	**0.63**	0.70	0.67	**0.62**	0.70	**0.66**
Biel	0.13	0.28	0.17	0.06	0.20	0.09

Table 4: Weighted and unweighted precision, recall and f-measure for all systems. Best performance per metric and dataset shown in bold.

Acknowledgements

The task organizers wish to thank Maria Giannoudaki and Maria Vomva for the editing of the hand-crafted grammars used in this evaluation task. The authors would like to thank the anonymous reviewer for the valuable comments and suggestions to improve the quality of the paper. This work has been partially funded by the SpeDial and PortDial projects, supported by the EU Seventh Framework Programme (FP7), with grant number 611396 and 296170 respectively.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A pilot on semantic textual similarity. In *Proceedings* *of the First Joint Conference on Lexical and Computational Semantics*, pages 385–393.

Frederic Bechet, Benoit Favre, Alexis Nasr, and Mathieu Morey. 2014. Retrieving the syntactic structure of erroneous ASR transcriptions for open-domain spoken language understanding. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4125–4129.

Yonatan Bisk and Julia Hockenmaier. 2012. Simple robust grammar induction with combinatory categorial grammars. In *Proceedings of the 26th Conference on Artificial Intelligence*, pages 1643–1649.

Bart Cramer. 2007. Limitations of current grammar induction algorithms. In *Proceedings of the 45th annual meeting of the ACL: Student Research Workshop*, pages 43–48.

Katerina T. Frantzi and Sophia Ananiadou. 1997. Automatic term recognition using contextual cues. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 41–46.

Spiros Georgiladakis, Christina Unger, Elias Iosif, Sebastian Walter, Philipp Cimiano, Euripides Petrakis, and Alexandros Potamianos. 2014. Fusion of knowledge-based and data-driven approaches to grammar induction. In *Proceedings of Interspeech (accepted)*.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Elias Iosif and Alexandros Potamianos. 2007. A soft-clustering algorithm for automatic induction of semantic classes. In *Proceedings of Interspeech*, pages 1609–1612.

Elias Iosif, Athanasios Tegos, Apostolos Pangos, Eric Fosler-Lussier, and Alexandros Potamianos. 2006. Unsupervised combination of metrics for semantic class induction. In *Proceedings of the International Workshop on Spoken Language Technology (SLT)*, pages 86–89.

Ioannis Klasinas, Alexandros Potamianos, Elias Iosif, Spiros Georgiladakis, and Gianluca Mameli. 2013. Web data harvesting for speech understanding grammar induction. In *Proceedings of Interspeech*, pages 2733–2737.

Jonas Kuhn. 2004. Experiments in parallel-text based grammar induction. In *Proceedings of the 42nd annual meeting of the ACL*, pages 470–477.

Helen M. Meng and Kai-chung Siu. 2002. Semi-automatic acquisition of semantic structures for understanding domain-specific natural language queries. *IEEE Transactions on Knowledge and Data Engineering*, 14(1):172–181.

Dmitrijs Milajevs and Matthew Purver. 2014. Investigating the contribution of distributional semantic information for dialogue act classification. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality*, pages 40–47.

Teruhisa Misu and Tatsuya Kawahara. 2006. A bootstrapping approach for developing language model of new spoken dialogue systems by selecting web texts. In *Proceedings of Interspeech*, pages 9–12.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the ACL*, pages 311–318.

Andrew N. Pargellis, Eric Fosler-Lussier, Chin-Hui Lee, Alexandros Potamianos, and Augustine Tsai. 2004. Auto-induced semantic classes. *Speech Communication*, 43(3):183–203.

Elias Ponvert, Jason Baldridge, and Katrin Erk. 2011. Simple unsupervised grammar induction from raw text with cascaded finite state models. In *Proceedings of the 49th annual meeting of the ACL*, pages 1077–1086.

Portdial. 2014a. PortDial project, final report on automatic grammar induction and evaluation D3.3. Technical report, https://sites.google.com/site/portdial2/deliverables-publications.

Portdial. 2014b. PortDial project, free data deliverable D3.2. Technical report, https://sites.google.com/site/portdial2/deliverables-publications.

Aarne Ranta. 2004. Grammatical framework: A type-theoretical grammar formalism. *Journal of Functional Programming*, 14(2):145–189.

Ruhi Sarikaya. 2008. Rapid bootstrapping of statistical spoken dialogue systems. *Speech Communication*, 50(7):580–593.

Abhinav Sethy, Shrikanth S. Narayanan, and Bhuvana Ramabhadran. 2007. Data driven approach for language model adaptation using stepwise relative entropy minimization. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 177–180.

Ye-Yi Wang and Alex Acero. 2006. Rapid development of spoken language understanding grammars. *Speech Communication*, 48(3-4):390–416.