

# Fusion of knowledge-based and data-driven approaches to grammar induction

Spiros Georgiladakis<sup>1</sup>, Christina Unger<sup>2</sup>, Elias Iosif<sup>1</sup>,  
Sebastian Walter<sup>2</sup>, Philipp Cimiano<sup>2</sup>, Euripides Petrakis<sup>1</sup>, Alexandros Potamianos<sup>3</sup>

<sup>1</sup>School of Electronic & Computer Eng., Technical University of Crete, Chania, Greece

<sup>2</sup>Cognitive Interaction Technology - Center of Excellence (CITEC), Bielefeld University, Germany

<sup>3</sup>School of Electrical & Computer Eng., National Technical Univ. of Athens, Zografou, Greece

{spgeo, petrakis}@intelligence.tuc.gr, {cunger, swalter, cimiano}@cit-ec.uni-bielefeld.de,  
{iosife, potam}@telecom.tuc.gr

## Abstract

Using different sources of information for grammar induction results in grammars that vary in coverage and precision. Fusing such grammars with a strategy that exploits their strengths while minimizing their weaknesses is expected to produce grammars with superior performance. We focus on the fusion of grammars produced using a knowledge-based approach using lexicalized ontologies and a data-driven approach using semantic similarity clustering. We propose various algorithms for finding the mapping between the (non-terminal) rules generated by each grammar induction algorithm, followed by rule fusion. Three fusion approaches are investigated: early, mid and late fusion. Results show that late fusion provides the best relative F-measure performance improvement by 20%.

**Index Terms:** spoken dialogue systems, corpus-based grammar induction, ontology-based grammar induction, grammar fusion

## 1. Introduction

Spoken language understanding (SLU) for commercial dialogue systems is generally based on hand crafted grammars that need to be maintained by the developer in order to improve their coverage [1]. Automated or semi-automated acquisition of grammars is a crucial task for the reduction of the cost that accompanies such processes. Approaches to automatically inducing grammars can be broadly divided into two categories [2]: knowledge-based (or top-down) and data-driven (or bottom-up) algorithms.

Knowledge-based algorithms rely on the manual development of domain-specific grammars or lexica. Various sources of domain knowledge are available nowadays in the form of ontologies; such knowledge is increasingly being exploited in dialogue systems [3, 4, 5]. In addition, research on ontology lexica [6] explores how such domain knowledge can be connected with rich linguistic information. Grammars that are generated from ontology lexica thus often achieve high precision but suffer from limited coverage. In order to improve coverage, regular expressions and word/phrase order permutations are used, however often at the cost of overgeneralization. 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. Data-driven approaches, on the other hand, rely solely on corpora of transcribed utterances [7, 8] and are therefore easier to port across languages and domains. However, since the corpora serve as in-domain data, the success of data-driven approaches strongly relies on their size and availability.

The use of different sources of information results in induced grammars of variable quality. Approaches to improve their performance include combination algorithms such as the linear combination of classifiers and grammars [9] or alteration of the input sources in order to improve the coherence between them and prevent overgeneralization [10]. Algorithms have also been developed (i.e., the fuzzy grammar similarity algorithm [11]) that measure the degree of similarity between grammars. In [12], a method was proposed for the extraction and clustering of phrases (n-grams) from corpora, where the clustered phrases were considered to correspond to grammar rules. A phrase grammar-based language model (LM) was built, which was found to yield higher performance compared to a phrase-based LM for an end-to-end SDS application. In [13], n-grams were combined with Stochastic Context-Free Grammars (SCFGs) for language modeling. A similar approach was followed in [14], where a unified model was proposed for integrating CFGs and n-grams for speech recognition and SLU. Both of the proposed models were reported to achieve lower perplexity compared to a trigram LM.

In this paper, we investigate how different grammar induction approaches can be fused in order to 1) eliminate problems that may occur in one of the grammars, such as overgeneralization, 2) extend grammar coverage by unifying information from different sources, and 3) perform better than the separate grammars by taking advantage of the different sources of information, resulting in a grammar with superior performance.

We present three fusion strategies: 1) in *early fusion* we expand the resulting grammar of the top-down approach into a seed corpus for use in the bottom-up approach (in spirit close to [15]), 2) in *mid fusion* we instead extract a list of its grammar rule fragments in order to feed the induction of bottom-up grammar rules through its respective induction system, and 3) in *late fusion* we combine the resulting two grammars using different approaches.

## 2. Grammar Induction Approaches

We review two approaches for grammar induction: a *bottom-up approach* based on corpus statistics, and a knowledge-based *top-down approach*, building on ontology lexica. For more details on an initial version of the algorithms see [16].

### 2.1. The Bottom-Up Approach

The main advantage of the bottom-up approach is that it is language-agnostic, using raw corpora as the only input for

grammar induction [16]. The corpus is created by extracting a linguistically rich subset of automatically web-harvested data relevant to the domain of interest, selected using queries formulated from a bootstrap grammar [17]. The bottom-up induction method consists of two main modules, whose goal is the induction of 1) terminal concepts and 2) non-terminal grammar rules.

A concept can be defined as a set whose members share the same meaning with respect to the domain of interest. In a flight travel domain, e.g., a concept `CITY` can describe a set comprising of city names such as:

```
<CITY> = (Boston, New York, Atlanta, ...)
```

Members of such a set are defined as terminals, i.e., single or multi-word terms that populate the leafs of domain taxonomies. Concept specifications like the one above are also referred to as terminal rules. Concept induction is realised by estimating the semantic similarity between the terminal tokens (words) that constitute the corpus vocabulary. Word similarities can be estimated by a variety of similarity metrics [18, 19]. In this work, the distributional hypothesis of meaning (i.e., "similarity of context implies similarity of meaning" [20]) is adopted and the semantic similarity between two words is estimated as the *Manhattan-norm* of their respective bigram probability distributions of left and right contexts [8].

The second module builds upon terminal concepts in order to induce grammar (non-terminal) rules. An example of such a rule is the following (where square brackets group optional elements):

```
<FROMCITY> = ["depart" | "leave" | "fly"  
              ("from" | "out of") <CITY>
```

For non-terminal rule induction, every instance of concepts within the corpus is substituted with their concept label and candidate sentential fragments are identified for the induction of grammar rules using a rule-based classification methodology based on heuristic criteria [21]. Subsequently, similarity between the selected fragments and seeding grammar rules is calculated with respect to the *longest common substring (LCS)* similarity metric.

The core idea of the bottom-up grammar induction is that 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 enhancement. First, candidate fragments are extracted from a corpus (all n-grams with n ranging typically between two and five) and are pre-filtered based on their respective frequency. Then, the grammar rule is enhanced by: 1) Pruning the list of candidates by removing "junk" fragments that are poor candidates for enhancement. The fragment pruning algorithm uses a statistical model, trained using lexical, syntactic and semantic features. 2) Ranking the candidates using a similarity metric in order to select the most appropriate. The similarity is estimated using lexical information. For more details see [21].

## 2.2. The Top-Down Approach

Ontology lexica can capture possible lexicalizations of ontology elements (classes, properties, and individuals) in several languages. They provide a compact declarative representation of syntactic and semantic aspects of lexical items, usually specifying the meaning of those items by pointing to a specific ontology element. Most importantly, they usually abstract from specific linguistic theories and grammar formalisms, therefore facilitating the construction of lexica by non-experts.

In order to exploit the linguistic knowledge captured in ontology lexica for spoken dialogue systems, top-down grammar induction implements a procedure that automatically generates grammars from ontology lexica, specifically generating ABNF grammars from lexica in *lemon* [22] format. The resulting grammars encompass semantic representations aligned with the underlying domain knowledge. Given a flight travel ontology, an example of a lexical entry for the noun "city" can be expressed by means of a pattern macro [23] as:

```
ClassNoun("city",ontology:City)  
  with plural "cities"
```

This specifies an entry that refers to the ontology class `City`, has part of speech `noun`, the singular, canonical form "city", and the plural form "cities".

In order to map an ontology lexicon to an ABNF grammar, first relevant information needs to be extracted from the lexicon. For common nouns such as the above, this mainly comprises of singular and plural forms. Then, based on the part of speech, a corresponding grammar template is instantiated. For common nouns, e.g., the template looks as follows (where slots are marked with boldface, `$DET_SG` extends to determiners such "a", "the", "each", and `$DET_PL` extends to determiners like "all"):

```
<reference_NP> = <DET_SG> singular  
                | [ <DET_PL> ] plural
```

Instantiating this template with information extracted from the entry "city" yields the following grammar fragment, which captures noun phrases such as "a city", "the city", "all cities", "cities", etc.:

```
<City_NP> = <DET_SG> city | [ <DET_PL> ] cities
```

Similarly, for individuals the lexicon would specify name entries, such as the following one:

```
Name("Boston",ontology:Boston)
```

Since `Boston` is an individual of type `City`, the corresponding grammar template extends the `City` noun phrase:

```
<type_NP> = form
```

Instantiated, this yields the following fragment:

```
<City_NP> = Boston ;
```

Ontology properties are often lexicalized as verbs. An example of such an entry is the following one:

```
StateVerb("depart",ontology:departureCity,  
          subjofProp = Subject,  
          objOfProp  = PrepositionalObject("from"))
```

It specifies an entry for the verb "to depart", which refers to the ontology property `departureCity`, where the domain of this property chain corresponds to the subject argument and the range corresponds to an object argument marked by the preposition "from". The corresponding grammar template specifies sentence fragments for all singular and plural forms of the verb:

```
<S> = <subject_NP_SG> formSG <object_NP>  
      | <subject_NP_PL> formPL <object_NP>
```

Instantiating this template with information from the above verb entry yields the following grammar fragment, covering sentences such as "the flight departs from Boston".

```
<S> =  
  <Flight_NP_SG> departs from <City_NP>
```

```
| <Flight_NP_PL> depart from <City_NP>
| <Flight_NP_SG> departed from <City_NP>
| <Flight_NP_PL> departed from <City_NP>
```

### 3. Fusion Strategies

Each grammar induction approach has different strengths and weaknesses. The top-down approach generates grammars from lexicalizations of ontology concepts, thereby covering mostly domain-specific vocabulary. For domain-independent vocabulary it has to rely on hand-crafted grammar modules and thus often lacks coverage. Furthermore, it usually fails with grammatically incorrect and fragmentary utterances. The bottom-up approach is language agnostic, relying on a raw corpus without the requirement of costly resources such as ontology lexica, however it is not fully unsupervised since seeding examples of grammar rules are needed. Also, the quality of the generated grammar is significantly affected by the richness of the seed corpus. Therefore, we aim at developing fusion strategies that combine both approaches resulting in grammars that improve the accuracy of the bottom-up and at the same time increase the coverage of the top-down grammar.

To this end we investigate three different strategies. In *early fusion*, we are expanding the top-down grammar creating a corpus that is used as input to the bottom-up grammar induction approach. The *mid fusion* strategy uses a list of grammar rule fragments from the top-down grammar as input for the bottom-up approach to further enhance its grammar. Finally, the *late fusion* strategy tries to combine the results of the two induction approaches into a new grammar.

Assume the following top-down induced grammar example:

```
<Location_NP> = Boston | Miami | Atlanta
<StopOver_NP> = some stops in <Location_NP>
```

and the following bottom-up induced grammar example:

```
<CITY> = Boston | Denver
<STATE> = Miami | Hawaii | Alaska
<STOPCITY> = stopover in (<CITY> | <STATE>)
```

In the following sections we discuss the different strategies in detail using the above grammar rules as examples.

#### 3.1. Early Fusion

In early fusion, the top-down grammar is expanded yielding all utterances it covers. The resulting top-down corpus is then used for 1) induction and 2) enhancement of the bottom-up grammar. In the first case, the top-down corpus is combined with the bottom-up corpus and their union is used for the induction of bottom-up grammar rules. In the second case, the top-down corpus serves as the seed corpus for the enhancement of already induced bottom-up grammar rules, i.e., seeding the bottom-up induction with the top-down corpus is done in addition to the usual bottom-up grammar induction.

Using the above examples, a relevant top-down corpus (after the substitution of its concepts) would be:

```
some stops in <CITY>
some stops in <STATE>
some stops in Atlanta
```

resulting in the enhancement of the bottom-up rule:

```
<STOPCITY> = stopover in (<CITY> | <STATE>) |
             some stops in (<CITY> | <STATE>)
```

#### 3.2. Mid Fusion

The mid-level fusion algorithm combines the two grammars at the candidate phrase fragment level, rather than at the corpus level. Specifically, the top-down grammar rules are expanded into phrase fragments that are used for bottom-up grammar 1) induction, i.e., merging them with the corpus-extracted fragments, and 2) enhancement, i.e., using the list as candidates for enhancement of an already bottom-up induced grammar. The fragments of the above top-down grammar would be (in this case only one):

```
some stops in <Location_NP>
```

resulting in the enhancement of the rule STOPCITY with

```
<STOPCITY> = stopover in (<CITY> | <STATE>) |
             some stops in <Location_NP>
```

#### 3.3. Late Fusion

In the late fusion strategy, both grammar induction processes run independently and the resulting grammars are merged. We investigate three different approaches for merging. The first one consists in a simple union of the two grammars. Fusing the above example grammars would thus yield the following union:

```
<Location_NP> = Boston | Miami | Atlanta
<StopOver_NP> = some stops in <Location_NP>
<CITY> = Boston | Denver
<STATE> = Miami | Hawaii | Alaska
<STOPCITY> = stopover in (<CITY> | <STATE>)
```

The other two fusion techniques take the correspondence of grammar rules into account by following a rule-based mapping strategy. Rule mappings are in general not one-to-one and are hard to determine automatically. To this end, we realised augmentation of the bottom-up grammar by matching its grammar rules with grammar rules of the top-down grammar in a many-to-one mapping and subsequently appended the rules of the latter to the best matching rule of the former, and vice versa. The mapping was done using the *Levenshtein* distance metric.

For example, augmenting the above bottom-up grammar with the top-down grammar yields the following grammar (assuming that the bottom-up rule STOPCITY is matched with the top-down rule StopOver\_NP):

```
<STOPCITY> = stopover in (<CITY> | <STATE>) |
             some stops in <Location_NP>
```

## 4. Evaluation

We evaluate the proposed fusion strategies on bottom-up and top-down grammars induced for the flight travel domain in English. Evaluation is done with respect to non-terminal rules, using a hand-crafted grammar that serves as gold standard.

The input for the bottom-up grammar induction approach is a web-harvested corpus comprising of 17,564 sentences. A bootstrap grammar was used to generate queries in order to retrieve web documents that were filtered as described in detail in [17] for the corpus creation. The bottom-up grammar was induced using the above corpus by bootstrapping each rule with two grammar fragments and requesting ten enhancements for each. The input for the top-down grammar induction approach is the flight travel ontology developed in the PortDial project [24] and a corresponding hand-crafted ontology lexicon, comprising 67 lexicalizations of the most important ontology elements.

The resulting grammars consist of 59 rules and 234 grammar fragments for the bottom-up grammar and 141 rules and 731 grammar fragments for the top-down grammar. Also, a top-down corpus was generated from the top-down grammar expansion, comprising 6,017 sentences, used for early fusion.

#### 4.1. Method and Evaluation Measures

For *early* and *mid fusion* experiments we requested 10 enhancements per rule. Each experiment was repeated 10 times and the evaluation results display the average evaluation of the separate resulting grammars. Experiments conducted regarding *late fusion* used, supplementary to the simple fusion approaches, an intrinsic and extrinsic concept matching on both rule and fragment level to improve the automated rule mapping. The *intrinsic matching* takes into account the terminals in all possible expansions of the concept and the *extrinsic matching* relies on the context of those concepts in the grammar rules in which they occur. Concept matching is done prior to rule matching in order to improve the performance of the latter. In addition, grammar merging was realised with respect to 1) *rule augmentation*, i.e., appending each rule of the augmenting grammar to its best matching grammar rule of the augmented grammar, and 2) *fragment augmentation*, where each fragment of the augmenting grammar is appended to its best matching grammar rule of the augmented grammar regardless of the rule it belongs to.

#### 4.2. Results and Discussion

Evaluation of the input grammars and the fusion strategies with respect to the gold standard grammar in terms of Precision (Pr), Recall (Rc) and F-Measure (Fm) is presented in Table 1, including the number of fragments of the resulting fused grammar.

Table 1: Evaluation of fusion strategies.

Fusion	Grammar	Pr	Rc	Fm	Fragm.
-	Bottom-up (BU)	0.65	0.44	0.52	234
-	Top-down (TD)	<b>0.81</b>	0.18	0.30	731
Early	BU Induction	0.66	0.37	0.47	182
Early	BU Enhance.	0.63	0.44	0.52	262
Mid	BU Induction.	0.64	0.52	0.58	314
Mid	BU Enhance.	0.52	0.55	0.53	437
Late	Union	0.72	0.55	<b>0.63</b>	965
Late	BU Augm. Rule	0.27	0.46	0.34	965
Late	BU Augm. Frag.	0.58	<b>0.58</b>	0.58	509
Late	TD Augm. Rule	0.61	0.20	0.30	965
Late	TD Augm. Frag.	0.79	0.21	0.34	787

##### 4.2.1. Early Fusion

Early fusion provided relatively poor results. Using the top-down generated corpus directly for bottom-up grammar induction provided a small improvement in precision with a loss in recall compared to bottom-up grammar induction. This is probably due to the different size of the corpora used for the bottom-up induction (17,564 sentences compared to 6,017 of the top-down). Early fusion enhancement provided better results but also failed to improve on the bottom-up baseline. This can be attributed to the quality and richness of the top-down corpus that probably affected the overall quality of the merged corpora. Upon its union with the bottom-up grammar, we induced grammar that verges towards the baseline with F-measure of 0.47.

##### 4.2.2. Mid Fusion

Results are better than the baseline grammar when following the mid-fusion strategy, achieving an F-measure of 0.58 when inducing the grammar using the union of the corpus-extracted and top-down fragments. Following the bottom-up enhancement using top-down fragments, coverage of the grammar has improved increasing by 12,5% to 0.55 compared to the initial 0.44 of the baseline bottom-up grammar. Mid-fusion significantly improves recall but at a loss in precision.

##### 4.2.3. Late Fusion

In late fusion, simple union outperforms all other approaches achieving an F-measure of 0.63, which is the best result overall. Regarding the augmentation methods, fragment level matching performs best both in terms of precision and recall with regard to both bottom-up and top-down augmentation. Overall, bottom-up fragment-based augmentation performs the best, reaching an F-measure of 0.58 during simple fragment matching (followed by the intrinsic and extrinsic fragment matching with F-measure of 0.55). Rule level matching preserved the problem of overgeneralization created to some extent by the fact that the grammars differ in their structure and organisation.

The above experiments showed that an important problem is rule matching. Especially, fragments often failed to merge although they semantically belonged to the same rule. This is explained to some extent by the different structure of the two grammars. Also, erroneous terminal concept mapping had deleterious effect on the subsequent rule mapping and thus posed another problem for the merging of the grammars. Even with perfect terminal concept mapping, the fusion of the mostly syntactically driven top-down grammar rules with the lexico-semantic driven bottom-up rules remains a challenge.

## 5. Conclusions

In this paper, we investigated various fusion algorithms for grammar induction. In particular, we presented different techniques combining a top-down, knowledge-based approach with a bottom-up, corpus-based approach. Our results indicate a 20% relative improvement on the performance of the input grammars by taking a simple union. However, this does not take into consideration the similarity and possible overlap of grammar rules as it does not capture the fact that different rules may cover the same information, making it a rather coarse technique.

In order to remove noise introduced by incorrect rule mappings we followed different strategies and created superior performing grammars by using bottom-up fragment-based augmentation. Although having a slightly inferior performance with respect to simple union, the problem of overgeneralization and rule matching is avoided.

## 6. Acknowledgements

This work has been partially funded by the PortDial ("Language Resources for Portable Multilingual Spoken Dialog Systems") and SpeDial ("Machine-Aided Methods for Spoken Dialogue System Enhancement and Customization for Call-Center Applications") projects supported by the EU Seventh Framework Programme (FP7), grant numbers 296170 and 611396 respectively. We would like to thank Ioannis Klasinas, Georgia Athanasopoulou and Maria Vomva for their constructive discussions and help with the experiments.

## 7. References

- [1] E. Barnard, A. Halberstadt, C. Kotelly, and M. Phillips, "A consistent approach to designing spoken-dialog systems," in *Proc. of Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 1999.
- [2] Y. Y. Wang and A. Acero, "Rapid development of spoken language understanding grammars." *Speech Communication*, vol. 48, no. 3-4, pp. 390–416, 2006.
- [3] D. Milward and M. Beveridge, "Ontology-based dialogue systems," in *3rd Workshop on Knowledge and Reasoning in Practical Dialogue Systems – 18th International Joint Conference on Artificial Intelligence*, 2003.
- [4] A. Flycht-Eriksson, "Design and use of ontologies in information-providing dialogue systems," Ph.D. dissertation, School of Engineering, Linköping University, 2004.
- [5] J. P. Pardal, "Dynamic use of ontologies in dialogue systems," in *Proc. of the NAACL-HLT 2007 Doctoral Consortium*. ACL, 2007, pp. 25–28.
- [6] L. Prévot, C. Huang, N. Calzolari, A. Gangemi, A. Lenci, and A. Oltramari, "Ontology and the lexicon: A multidisciplinary perspective," in *Ontology and the Lexicon: A Natural Language Processing Perspective*. Cambridge University Press, 2010, pp. 3–24.
- [7] H. M. Meng and K. Siu, "Semiautomatic acquisition of semantic structures for understanding domain-specific natural language queries." *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 1, pp. 172–181, 2002.
- [8] A. N. Pargellis, E. Fosler-Lussier, C. Lee, A. Potamianos, and A. Tsai, "Auto-induced semantic classes." *Speech Communication*, vol. 43, no. 3, pp. 183–203, 2004.
- [9] A. Przepirkowski, M. Marcinczuk, and L. Degrski, "Dealing with small, noisy and imbalanced data." in *TSD*, ser. Lecture Notes in Computer Science, P. Sojka, A. Hork, I. Kopecek, and K. Pala, Eds., vol. 5246. Springer, 2008, pp. 169–176.
- [10] N. M. Sharef, T. Martin, and Y. Shen, "Minimal combination for incremental grammar fragment learning." in *Proc. International Conference on Fuzzy Systems Association/European Society for Fuzzy Logic and Technology (IFSAC/EUSFLAT)*, 2009, pp. 909–914.
- [11] G. Petasis, G. Paliouras, C. D. Spyropoulos, and C. Halatsis, "Eg-GRIDS: Context-Free Grammatical Inference from Positive Examples Using Genetic Search," in *Grammatical Inference: Algorithms and Applications*, in *Proc. of the 7th International Colloquium on Grammatical Inference (ICGI 2004)*, ser. Lecture Notes in Computer Science, G. Paliouras and Y. Sakakibara, Eds., vol. 3264. Athens, Greece: Springer Berlin / Heidelberg, October 11–13 2004, pp. 223–234.
- [12] G. Riccardi and S. Bangalore, "Automatic acquisition of phrase grammars for stochastic language modeling," in *Proc. of the 6th Workshop on Very Large Corpora.*, 1998.
- [13] J. M. Benedi and J. A. Snchez, "Combination of n-grams and stochastic context-free grammars for language modeling," in *Proc. of Proceedings of the 18th Conference on Computational Linguistics (COLING)*, vol. 1, 2000, pp. 55–61.
- [14] Y. Y. Wang, M. Mahajan, and X. Huang, "A unified context-free grammar and n-gram model for spoken language processing," in *IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 3, 2000.
- [15] S. P. Caskey, E. Story, and R. Pieraccini, "Interactive grammar inference with finite state transducers," in *Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2003, pp. 572–575.
- [16] PortDial Project, "D3.1 Free Data Deliverable," 2013, <https://sites.google.com/site/portdial2/deliverables-publications>.
- [17] I. Klasinas, A. Potamianos, E. Iosif, S. Georgiladakis, and G. Mameli, "Web data harvesting for speech understanding grammar induction." in *Interspeech*, 2013, pp. 2733–2737.
- [18] E. Iosif and A. Potamianos, "Similarity computation using semantic networks created from web-harvested data," *Natural Language Engineering*, vol. FirstView, pp. 1–31, 2 2014.
- [19] E. Iosif, "Network-based distributional semantic models," Ph.D. dissertation, School of Electronic and Computer Engineering, Technical University of Crete, 2013.
- [20] Z. Harris, "Distributional structure," *Word*, vol. 10, no. 23, pp. 146–162, 1954.
- [21] G. Athanasopoulou, I. Klasinas, E. Iosif, and A. Potamianos, "Using lexical, syntactic and semantic features for non-terminal grammar rule induction in spoken dialogue systems," in *Interspeech*, (submitted) 2014.
- [22] J. McCrae, G. A. de Cea, P. Buitelaar, P. Cimiano, T. Declerck, A. Gomez-Perez, J. Garcia, L. Hollink, E. Montiel-Ponsoda, and D. Spohr, "Interchanging lexical resources on the semantic web," *Language Resources and Evaluation*, vol. 46, no. 4, pp. 701–719, 2012.
- [23] J. McCrae and C. Unger, "Design patterns for engineering the ontology-lexicon interface," in *Towards the Multilingual Semantic Web*, P. Buitelaar and P. Cimiano, Eds. Springer, (to appear) 2014.
- [24] PortDial Project, "D2.1 Free Data Deliverable," 2013, <https://sites.google.com/site/portdial2/deliverables-publications>.