Graph Databases for Designing High-Performance Speech Recognition Grammars

Maria Di Maro
Università degli Studi di Napoli Federico II
mdimaro17@gmail.com

Marco Valentino
Università degli Studi di Napoli Federico II
m.valentino91@gmail.com

Anna Riccio
Università degli Studi di Napoli 'L'Orientale'
ariccio@unior.it

Antonio Origlia
Università degli Studi di Padova
antonio.origlia@unipd.it

Abstract

The present paper reports on the advantages of using graph databases in the development of dynamic language models in Spoken Language Understanding applications, such as spoken dialogue systems. First of all, we introduce Neo4J graph databases and, specifically, MultiWordNet-Extended, a graph representing linguistic knowledge. After this first overview, we show how information included in graphs can be used in speech recognition grammars to automatically extend a generic rule structure. This can be the case of linguistic elements, such as synonyms, hypernyms, meronyms and phonological neighbours, which are semantically or structurally related to each other in our mental lexicon. In all the AI based approaches depending on a training process using large and representative corpora, the probability to correctly predict the creativity a speaker can perform in using language and posing questions is lower than expected. Trying to capture most of the possible words and expressions a speaker could use is extremely necessary, but even an empirical, finite collection of cases could not be enough. For this reason, the use of our tool appears as an appealing solution, capable of including many pieces of information. In addition, we used the proposed tool to develop a spoken dialogue system for museums and the preliminary results are shown and discussed in this paper.

1 Introduction

While research on Natural Language Understanding is still investigating how to reliably interpret unconstrained user utterances, practical applications that are now common on mobile devices, like virtual assistants, heavily rely on utterance templates to provide their services. Such language models can be dynamically loaded depending on the situation so that the speech recognition engine becomes biased towards the set of utterances the underlying dialogue system is able to manage. For relatively small and well-defined dialogue domains, using this kind of language model appears to be a practical choice for application developers. The problems posed by language variability, however, do not only impact the functionality of spoken dialogue systems at run time: developing grammar-based language models can be a time-consuming and error-prone task by itself. Taking advantage of structured linguistic knowledge to overcome this aspect of dialogue systems design has led, in the past, to the use of linguistic ontologies to automatically expand the set of terms accepted by the speech recognition system (Milward and Beveridge, 2003). Using lexical ontologies to represent the knowledge a machine has to process is something which has been investigated in early years: WordNet (Miller, 1993), (Fellbaum, 1998) was used in machine translation (Knight, 1993), information extraction (Burke et al., 1995), automatic text summarisation (Chaves, 2001) and domain-specific dialogue systems management (Snae and Bruckner, 2008). In previous works, the preferred way of using ontological knowledge in dialogue systems appears to be based on the use of explicit reference to the classes defined in the taxonomy. This, however, implies that ontologies supporting the dialogue domain must already exist or be constructed before the system can take advantage from it. Modern approaches to data representation, however, use powerful querying languages to extract knowledge that is not explicitly structured in the ontological organisation by the presence of dedicated classes. Moreover, it appears that, while the interest towards using ontologies in dialogue systems is well-established, there has been a less significant effort towards the definition of a common way to merge grammar definitions supporting ontological expansions. In this paper, we present a formal language to describe ontologically-enriched language models. This language is obtained by expanding the W3C Speech Recognition Grammar Specification (SRGS) XML standard (Hunt and McGlashan, 2003) with an item dedicated to queries directed towards knowledge bases returning lists of words. In our work, we generalise the use of ontologies by proposing the integration with a graph database, which can represent ontologies as well as other forms of data representation based on objects and relationships among objects. The paper is organised as follows: in Section 2 we describe the Neo4J graph database (Webber, 2012), which is the system we consider for knowledge representation in our work, and highlight the advantages it poses for dialogue systems support. In the same Section, we also describe a specific Neo4J database hosting different types of linguistic information ranging from morphology to phonology. In Section 3, we describe the extended SRGS language we designed while, in Section 4, we present some use cases of interest. In the closing Section 5, we show the results we got in testing a spoken dialogue system using the designed grammars, in order to prove its quality.

2 Neo4J Graph Database

Neo4J[1] is a no-SQL database adopting a graph-based representation of data. Its approach is very similar to the Resource Description Framework (RDF), which is often adopted for Linked Open Data. Neo4J and RDF representations can be both graphically described using nodes and arcs but there is a fundamental difference in the focus put on graph elements. While RDF is edge-centred, graph databases are node-centred. Furthermore, graph databases also explicitly distinguish *properties* from relationships and both nodes and relationships can be specified with labels, which are close to the concept of class in ontologies. Another important difference lies in the way the two approaches provide access to the hosted data. RDF can be queried using inference rules to extract implicit information from the explicit relationships. Graph databases are more suited to graph traversing and path finding. Choosing one over the other may depend on the application but, in the domain of grammars for dialogue systems, the more natural way of representing data is an advantage to keep the specified domains clear. Neo4J, in particular, has developed a powerful, and yet clear, querying language - Cypher - that makes heavy use of ASCII graphical characters to build its syntax. Among the most important ones, round brackets are used to specify nodes, square brackets for relationship labels, ASCII arrows to link nodes specifying arcs orientations (if necessary), and curly brackets to specify for properties. Being ontologies, it is straightforward to think of the linguistic organisation proposed in Wordnets as graphs. In our work, we adopt a Neo4J conversion of a wordnet for Italian that has been extended to include phonological data. We will present how Cypher can be used to extend grammar templates using linguistic and domain knowledge.

2.1 MutiWordNet-Extended

MultiWordNet-Extended (MWN-E) (Origlia et al., 2017) is a Neo4J database representing part of the Italian lexicon. It combines linguistic resources, specifically MultiWordNet (Pianta et al., 2002), Morph-it (Zanchetta and Baroni, 2005), TreeTagger (Schmid et al., 2007; Schmid, 2013) and the ISTC pronunciation dictionary (Cosi et al., 2001). The resulting database is a collection of data concerning semantic relations between lemmas. Relationship types cover both morpho-syntactic and phonological aspects. All the words contained in the database are taken from MultiWordNet and stored as nodes. They are given a morpho-syntactic label, such as NOUN, VERB, ADVERB or ADJECTIVE and are linked to SYNSET nodes by means of the semantic relationship BELONGS_TO. Synsets are then linked to one another with

[1]https://neo4j.com/product/

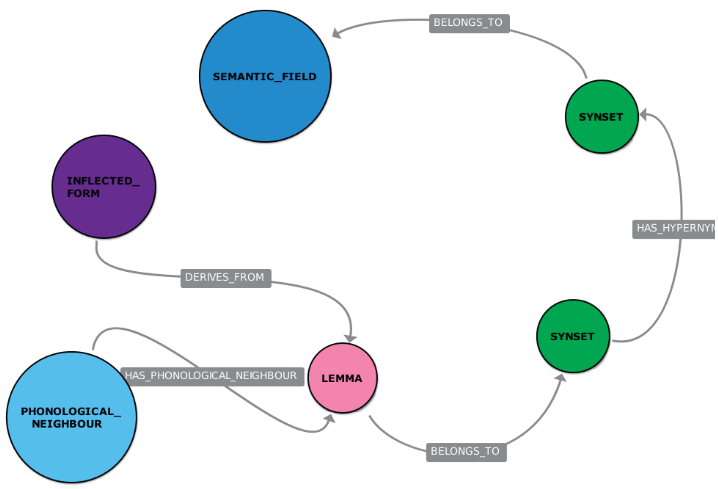

Figure 1: Nodes and relationships in MWN-E

an hypernymic relation and to SEMANTIC_FIELD nodes with a BELONGS_TO connection. By explicating this relationships, a semantic annotation is carried out. MultiWordNet, by itself, is not a complete representation of the language knowledge. For example, including the most common inflected forms and their phonological transcription makes it possible to use the same database to combine phonological and semantic information. To extend MultiWordNet this way, a procedure to import morpho-syntactic data using Morph-it and TreeTagger was adopted to add data like gender, number and inflection types for derived forms. Also, the ISTC pronunciation dictionary, was used to import phonological transcriptions in SAMPA format, including syllabification and accent position. Inflected forms are represented as new nodes, linked to their lemmas using a DERIVES_FROM relationship. Also, phonological neighbourhoods were computed and represented using HAS_PHONOLOGICAL_NEIGHBOUR connections. This kind of connection has a property describing the phonological distance (i.e. the Minimum Edit Distance between the SAMPA trascriptions) to distinguish homophones from actual neighbours. Figure 1 shows the general organisation of the database. Using MWN-E can be essential for a developer to easily create a language model including sets of words with a connected meaning, without having to write them manually. An example may be the need of encompassing the synonyms of a word used in a particular semantic field, as shown in the following query (1).

```
MATCH (n:NOUN {word:'automobile'})-[:BELONGS_TO]->(m:SYNSET)<-[:BELONGS_TO]-(k:NOUN),
(m)-[:BELONGS_TO]->(j:SEMANTIC_FIELD {name:'Tourism'})
RETURN distinct n.word, k.word
```
(1)

With this query we want to look for synonyms of the word *automobile* ('car') belonging to the semantic field 'Tourism'. Synonyms that are found are *auto*, *autovettura*, *vettura*, and *macchina*. They are part of a synset, which we can refer to in a grammar, without explicitly specify each lemma. In the next sections, it will be exemplified how we integrated Neo4J queries within a standard language for speech recognition grammars and we will show some examples of spoken dialogue systems taking advantage of this knowledge.

3 Speech Recognition Grammars and our extension proposal

Language data are formalised in order to let a machine be able to understand natural language and to actively produce it in giving answers, as far as a spoken dialogue system is concerned. This data can be organised in a grammar. The W3C standard we refer to in this paper is the Speech Recognition Grammar Specification (SRGS). This standard has been developed to allow Automatic Speech Recognition (ASR) engines to output the semantic interpretation of the matched pattern instead of the raw transcription. This is an important advantage for spoken dialogue systems as they can instruct ASR modules to *expect* specific word patterns and to present a structured interpretation of the obtained input to be provided to the dialogue manager. This results in reduced latency, as linguistic analysis chains working on raw strings are avoided, and in a more definite separation between dialogue management and input management. A speech recognition grammar $G = \{R_1, R_2, ...R_n\}$ is a finite set of rules. Each R_i generates a set A_i of utterances associated to semantic labels S_i so $R_i \vdash A_i, S_i$. Given a collection of semantic relations $X = \{x_1, x_2...x_n\}$, the tool described in this paper aims at expanding each R_i in a grammar G, according to $x_i \in X$, in order to produce a new grammar G' where the set of generated utterances A_i is included in A'_i. More formally:

$$T(G, X) = G' \mid \forall R'_i \in G', \quad R'_i \vdash A'_i, S_i \rightarrow A_i \subseteq A'_i$$

In SRGS, XML is used to describe what the ASR engine should expect. It is, then, possible to specify how the matched pattern should be returned to the dialogue manager in terms of semantic interpretation by using the W3C Semantic Markup Language (SML) (Van Tichelen and Burke, 2007). When combined, these two languages allow a dialogue manager to instruct and ASR engine to provide the semantic interpretation corresponding to actions and parameters it can handle in its current state. Listing all words accepted in a specific position, however, can be time-consuming for the dialogue designer. In the following example, suppose we want to match the regular expression pattern “dipingere .* quadro" (‘do .* painting’) against a generic output “PAINT" the dialogue manager can handle. Suppose we also want to accept all possible synonyms of the verb *dipingere* (‘to paint’) and of the noun *quadro* (‘painting’). Using SRGS, we would obtain the following

```
<rule id="Query" scope="public">
   <tag> out.QueryType="NONE"</tag>
   ...
   <ruleref uri="#PAINT" /> <tag> out.QueryType=rules.PAINT</tag>
   ...
</rule>

<rule id="PAINT">
   <ruleref special="GARBAGE"/>
   <one-of>
      ...
      <item>dipingere</item>
      <item>raffigurare</item>
      ...
   </one-of>
   <ruleref special="GARBAGE"/>
   <one-of>
      ...
      <item>quadro</item>
      <item>dipinto</item>
      ...
   </one-of>
</rule>
```

Lists of synonym words, however, can be obtained by querying the MWN-E database. We therefore use an attribute *query* for the generic tag *item* to indicate that the item should be substituted with a list of alternative words, thus obtaining the more compact representation

```
<rule id="Query" scope="public">
   <tag> out.QueryType="NONE"</tag>
   ...
   <ruleref uri="#PAINT" /> <tag> out.QueryType=rules.PAINT</tag>
   ...
</rule>

<rule id="PAINT">
   <ruleref special="GARBAGE"/>
   <item query="MATCH ({word:"dipingere"})-[:BELONGS_TO]->()<-[:BELONGS_TO]-(m) RETURN m.
       word">dipingere</item>
```

```
    <ruleref special="GARBAGE"/>
    <item query="MATCH ({word:"quadro"})-[:BELONGS_TO]->()<-[:BELONGS_TO]-(m) RETURN m.word">
        quadro</item>
</rule>
```

In this example, the Cypher query identifies all the nodes that have a BELONGS_TO relationship with the same synset the node having the value *dipingere* is part of, thus obtaining synonyms. Of course, this is not anymore an SRGS compliant XML document. A dialogue manager using these SRGS *templates* should be equipped with a simple software module to obtain an SRGS compliant XML document that can be passed to an ASR engine.

4 Case Study: Spoken Dialogue System for Cultural Heritage

Imagine you want to create a spoken dialogue system able to understand questions about artworks contained in a museum. Instead of writing all the possible words occurring in a particular syntactic structure to express a concept, we can include queries in our grammars. Speech recognition grammars were built in a way that made them suitable to represent structurally different questions expressing the same meaning. In the example (1), about materials, the rule contains different synonyms of the verb *utilizzare* ('to utilise'), such as *adoperare* ('to employ'), *usare* ('to use'), *impiegare* ('to use') and their inflected forms. Similarly, in the example (2), the synonyms of the word *periodo* ('period'), such as *contesto* ('context'), *epoca* ('epoch'), *secolo* ('century'), *milieu* and the inflected forms of the verb *vivere* ('to live') were encapsulated in the same rule, enabling the system to automatically understand different expressions of the same question:

(1) *Quali materiali utilizza l'artista?* 'Which materials did the artist use?'

(2) *In che periodo è vissuto l'artista?* 'In which period did the artist live?'

For example, the question in example (1) can be modelled as follows:

```
<item>
    <item>
        <one-of>
            <item>che</item>
            <item>quali</item>
        </one-of>
    </item>
    <ruleref special="GARBAGE"/>
    <item>materiali</item>
    <ruleref special="GARBAGE"/>
    <item query="MATCH (n:VERB {word:'usare'})-[:BELONGS_TO]->(s:SYNSET)
            MATCH (:SEMANTIC_FIELD {name: 'Factotum'})<-[:BELONGS_TO]-(s)<-[:BELONGS_TO]-(k:
                VERB)<-[:DERIVES_FROM]-(q:VERB) RETURN distinct q.word">usare</item>
    <ruleref special="GARBAGE"/>
    <item>artista</item>
</item>
```

By simply including a query like **MATCH (n:VERB {word:'usare'})-[:BELONGS_TO]-> (s:SYNSET), MATCH (:SEMANTIC_FIELD {name: 'Factotum'})<-[:BELONGS_TO]-(s)<-[:BELONGS_TO]-(k:VERB)<-[:DERIVES_FROM]-(q:VERB) RETURN distinct q.word**, it was possible to get an extended grammar encompassing all the synonyms and the inflected forms of the verb *usare* ('to use'), in order to be able to correctly process different expression of the same question. The obtained extended grammar is shown in the below xml code:

```
<rule>
    <item>
        <one-of>
            <item>usare</item>
            <item>uso</item>
            <item>usi</item>
            <item>usa</item>
```

```
            ...
            <item>utilizzare</item>
            ...
            <item>impiegare</item>
            ...
            <item>adoperare</item>
            ...
        </one-of>
    </item>
</rule>
```

Other content and formal information can be included by referring to MWN-E. As far as semantic aspects are concerned, we consider meronymic and hyperonymic relationships. Meronymy is concerned with part-whole relationships at a semantic level, meaning that entities are not only physically but also conceptually connected (Croft and Cruse, 2004). Speakers conceptualise reality and create specific significant associations between concepts, as a result that not each part of a whole can have relevance. Some entities are so relevant that can be used to refer to the whole they are part of. Part-whole associations are contained in the graph database MWN-E and including them can be useful to cover all the possible utterances a speaker can produce in a creative way. Suppose a speaker wants to ask the system a question on the painting "Tamara in a Green Bugatti" by Tamara de Lempicka, as in the example (3):

(3) *Chi è il personaggio al volante? = Chi è il personaggio nell'auto?*
'Who is the subject behind the wheel?' = 'Who is the subject in the car?'

This variation can be ruled by using a query, as in (2), including all the parts a car is made up of:

```
MATCH (n:NOUN{word:'auto'})-[:BELONGS_TO]->(l)-[:HAS_PART]->(m)-[:BELONGS_TO]-(g)<-
[:BELONGS_TO]-(p)<-[:BELONGS_TO]-(h)
MATCH (p:SYNSET)-[:BELONGS_TO]->(g:SEMANTIC_FIELD{name:'Mechanics'})
MATCH (m:SYNSET)-[:BELONGS_TO]->(g:SEMANTIC_FIELD{name:'Mechanics'})
RETURN n.word, l.word, m.word, g.word, p.word, h.word
```
(2)

It is interesting to notice that meronymy is mainly concerned with conceptual relationships belonging to the same conceptual domain; for this reason, this kind of relation, as opposed to metaphors, can be better represented and formalised within an ontology.

Another important information to include is the relationship between general and specific words. Words with a general meaning, also known as hypernym, express concepts which are easily accessible to speakers (Feldman, 2013). For this reason, these words correspond to the basic level of categorisation (Rosch, 1976). Hypernymic words are, therefore, more frequently used, even when we want to refer to something which can be linguistically specified. Supposing we want to model the lexical variety concerning a particular frame, such as ART, we surely need specific terms, but being able to automatically include their hypernym is important to avoid misrecognitions. For instance, *pittore* ('painter) is the exact term used to refer to the author of a painting, but the basic term *artista* ('artist') can also be used to refer to the same concept. These terms are indeed related by an hypernymic relation in our database. Another example concerning hyperonymy can refer to the relation between *tela* ('canvas') and *quadro* ('painting'), which are interchangeable in some contexts, such as *Chi è l'autore di questa tela?* ('Who is the author of this canvas?') and *Chi è l'autore di questo quadro?* ('Who is the author of this painting'). In MWN-E they are related to each other by means of an hypernymic relationship, in that *tela* is an hyponym of *quadro*. For this reason, this information can be also included in our grammars.

Phonological neighbourhoods can, instead, be useful to work with words that *sound* similar. In the case of logopedic applications, for example, phonological neighbours can be used to develop dialogue systems to test the capability of distinguishing consonantic traits, in people with speech disorders. Since the speech recogniser could misunderstand some similar words, phonological neighbours can be here used to improve the quality of the recognition, as we will see in the next Section.

	Artist's name	Artist's place	Artist's time	Materials	Techniques	Painting's Time	Style	Iconography	Painting's place	Painting's name	Aim/Function	Dimension	Realisation time	Legends	Commission	Elements	Economic value
Artist's name	14																
Artist's place	1	19															
Artist's time			15														
Materials				12													1
Techniques					16					2							
Painting's time			3			17										1	
Style							18								1		1
Iconography								15	1		1					1	
Painting's place									16								
Painting's name	1					1			1	15							
Aim/Function	1										14						
Dimension												16					1
Realisation time													16				
Legends				1										13			
Commission															12		
Elements																12	
Economic value																	10
NC	3	1	2	7	4	2	2	5	2	3	5	4	4	7	7	6	7
%Correct Classification	70%	95%	75%	60%	80%	85%	90%	75%	80%	75%	70%	80%	80%	65%	60%	60%	50%
%Incorrect Classification	30%	5%	25%	40%	20%	15%	10%	25%	20%	25%	30%	20%	20%	35%	40%	40%	50%

Figure 2: Confusion Matrix showing the classification results

5 Results and Discussions

To verify the quality of the described extended grammars, we tested them in the pipeline of a spoken dialogue system. Specifically, the test intends to verify if the system is able to understand the belonging class of posed questions. The suggested classes were *Artist's name*, *Artist's place*, *Artist's time*, *Materials*, *Techniques*, *Painting's time*, *Style*, *Iconography*, *Painting's place*, *Painting's name*, *Aim/Function*, *Dimension*, *Realisation time*, *Legends*, *Commission*, *Elements* and *Economic value*. 10 testers were shown two paintings and were requested to ask for information about them. In particular, they had to ask 2 questions per class, a simple and a more complex one. In total, we were able to analyse 340 questions in this test. This was important to prove that the system was able to understand the meaning of a question, in order to map the correct answer on the right semantic class. Identifying the correct class can be seen as a conceptual hinge between the received input and the output to be generated, and a well-formed grammar is the key part of this process. In order to verify the performance of the designed grammars an experimental system setup was built. A 3D scene, designed with Unreal Engine 4, was designed to show the considered paintings to the users. The Kinect 2.0 was applied as speech recogniser, adopting grammars modelled through the SRGS standard. The test that we carried out shows that, by using our grammar, the confusion between the classes is low (19 questions out of 340), as shown in the confusion matrix in Figure 2. This is easily possible, since the classes are well described by means of the automatic extension to all the semantic and grammatical relations of the few included words used to express a particular concept.

To get a deeper insight, we can divide the classes into three groups, according to their precision:

- 7 high-performance classes (in green): for the classes *Artist's Place*, *Techniques*, *Painting's Time*, *Style*, *Painting's Place*, *Dimension* and *Realisation Time* the classification was correct in the 80%-100% of the cases;

- 6 medium-performance classes (in blue): for the classes *Artist's Name*, *Artist's Time*, *Iconography*, *Painting's Name*, *Aim/Function* and *Legends* the classification was correct in the 65%-75% of the cases;

- 4 low-performance classes (in red): for the classes *Materials*, *Commission*, *Elements* and *Economic value* the classification was correct in the 50%-60% of the cases.

The modelled questions were mostly well classified by the system, but some questions (71 out of 340) were not recognised at all, because some of them were not included in the grammar, for further empirical collections are still needed. Most of the non-recognised questions belong to the last 4 low-performance classes. As a matter of fact, for these classes a lot of questions asked were not included in our grammars, for they were not prototypical. For example, the question *Ha un valore molto alto?* ('Does the painting have a high economic value?') was not recognised because not included, since the modelled questions were mostly wh-questions and not yes/no questions. Even though we were able to automatically include a lot of pieces of information using our tool, we still need to list all the possible syntactic structures. Further implementations of the grammars expect the possibility to structure the SRGS in an even more general way, in that we could automatically include FrameNet information in it (Ruppenhofer et al., 2010). In FrameNet each word is indeed semantically and syntactically described. The reference to a particular semantic frame could be used to include all the pieces of information expressed by the words used in that frame. Another problem of misrecognitions was caused by phonologically similar words. For example, *Che valore ha l'opera?* ('Which is the value of the artwork?') was understood by the speech recogniser as *Che colore adopera?* ('Which colour was used?'), since *valore-colore* and *opera-adopera* are phonological neighbours. Using the phonological information included in MWN-E will be necessary to avoid this kind of misrecognition in the future.

Despite the on-going improvements, we can assert that the concision of the grammar structure, clarified in the examples, and the rapidity of database reaction (less than a second) make clear the advantages of using this automatic module instead of exclusively relying on the manual one. While the expertise with the Cypher language is still an asset to guarantee the speed of compiling, the impossibility to think of any possible related word would consistently be time-consuming without the use of such database. Moreover, we think that the use of grammars is the most appropriate choice for specific domain applications, instead of using a general purpose ASR. Especially apps mostly take advantage from this approach, since their functions are well-defined. In such contexts, an ex post linguistic analysis on strings containing unpredictable information would be less convenient, considering that ASRs can be taught what to expect. As a result, our specific-purpose aimed application better relies on grammars whose performances are increased through an automatised extension module of analysis.

6 Conclusions

We presented an extension to the SRGS language for grammar models specification to integrate knowledge hosted by a graph database in dialogue management systems. We showed how Cypher queries can be used to generate lists of words complying to complex patterns involving morpho-syntactic and phonological constraints. The resulting SRGS documents provide a compact and easy-to-manage representation of the rules and can significantly reduce the time needed to design dialogue management. Future work will concentrate on using SRGS templates and the automatic conversion module to deploy virtual assistants for interactive cultural heritage tours.

Acknowledgements

This paper has been developed within the framework of CHROME (*Cultural Heritage Resources Orienting Multimodal Experience* - PRIN 2015 MIUR), an ongoing Italian project on technologies for cultural heritage. Antonio Origlia's work is supported by Veneto Region and European Social Fund (grant C92C16000250006).

References

Burke, R., K. Hammond, and J. Kozlovsky (1995). Knowledge-based information retrieval from semi-structured text. In *Working Notes from AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval*, pp. 19–24.

Chaves, R. P. (2001). Wordnet and automated text summarization. In *NLPRS*, pp. 109–116.

Cosi, P., F. Tesser, R. Gretter, C. Avesani, and M. Macon (2001). Festival speaks italian! *7th European Conference on Speech Communication and Technology*.

Croft, W. and A. Cruse (2004). *Cognitive Linguistics*. New York: Cambridge University Press.

Feldman, L. B. (2013). *Morphological Aspects of Language Processing*. Taylor & Francis.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Cambridge: MA: MIT.

Hunt, A. and S. McGlashan (2003). Speech recognition grammar specification version 1.0. Technical report, W3C.

Knight, K. (1993). Building a large ontology for machine translation. In *Proceedings of the workshop on Human Language Technology*, pp. 185–190. Association for Computational Linguistics.

Miller, G. A. (1993). Wordnet: A lexical database for english. *Communications of the ACM 38*(11), 39–41.

Milward, D. and M. Beveridge (2003). Ontology-based dialogue systems. In *Proc. 3rd Workshop on Knowledge and reasoning in practical dialogue systems (IJCAI03)*, pp. 9–18. Citeseer.

Origlia, A., G. Paci, and F. Cutugno (2017). Mwn-e: a graph database to merge morpho-syntactic and phonological data for italian. In *Proc. of Subsidia*, pp. to appear.

Pianta, E., L. Bentivogli, and C. Girardi (2002). *MultiWordNet: developing an aligned multilingual database*, pp. 293–302.

Rosch, E. (1976). Structural bases of typicality effects. *Journal of Experimental Psychology: Human Perception and Performance*, 491–502.

Ruppenhofer, J., M. Ellsworth, M. R. L. Petruck, C. R. Johnson, and J. Scheffczyk (2010). *FrameNet 2: Extended Theory and Practice*.

Schmid, H. (2013). Probabilistic part-of speech tagging using decision trees. In *New methods in language processing*, pp. 154. Routledge.

Schmid, H., M. Baroni, E. Zanchetta, and A. Stein (2007). The enriched treetagger system. In *proceedings of the EVALITA 2007 workshop*.

Snae, C. and M. Bruckner (2008). Foods: a food-oriented ontology-driven system. In *Digital Ecosystems and Technologies, 2008. DEST 2008. 2nd IEEE International Conference on*, pp. 168–176. IEEE.

Van Tichelen, L. and D. Burke (2007). Semantic interpretation for speech recognition version 1.0. Technical report, W3C.

Webber, J. (2012). A programmatic introduction to neo4j. In *Proceedings of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity*, SPLASH '12, New York, NY, USA, pp. 217–218. ACM.

Zanchetta, E. and M. Baroni (2005). Morph-it! a free corpus-based morphological resource for the italian language. In *PROCEEDINGS OF CORPUS LINGUISTICS*, Birmingham, UK. University of Birmingham.