

SMARTGRAMMAR: A DYNAMIC SPOKEN LANGUAGE UNDERSTANDING GRAMMAR FOR INFLECTIVE LANGUAGES

Vincenzo Catania¹ Raffaele Di Natale² Ylenia Cilano³ Antonio Rosario
Intilisano⁴ Daniela Panno⁵

¹DIEEI, University of Catania, Italy

²DIEEI, University of Catania, Italy

⁴DIEEI, University of Catania, Italy

³A-Tono Corporate, Italy

⁵DIEEI, University of Catania, Italy

ABSTRACT

Spoken language understanding (SLU) is a key requirement of spoken dialogue systems (SDS). The role of SLU parser is to robustly interpret the meanings of users' utterance using a hand-crafted grammar that is expensive to build. This task becomes even harder when the developer is creating a SLU grammar for inflectional languages due to the different conjugations and declensions. This causes long grammar definition files that are hard to structure and also to manage. In this paper, we propose a new and alternative method, called Smart Grammar to facilitate the development of speech enabled applications. This uses a morphological analyzer, in addition to the semantic parser, in order to convert each user utterance in the canonical form.

KEYWORDS

Spoken Language Understanding, Semantic Grammar, Spoken Dialog Systems

1. INTRODUCTION

In SLU systems, the semantic parser is based on hand-crafted grammar. These typically use context-free semantic rules to extract keywords or phrases to fill slots in semantic frames, examples are MIT's TINA [1], CMU's PHOENIX [2], and SRI's Gemini [3]. One of the problems is the requirement of intensive language engineering, a task that is time-consuming, error-prone and requires a significant amount of expertise [4]. The goal of this paper is to provide a new grammar definition format for the Phoenix Semantic Parser. This format, supported by our Smart Grammar tool, is very easy to write and to manage because it is able to automatically expand it making the grammar more rich and complete. This paper is organized as follows. A SLU Grammar is described in Section I, followed by the Phoenix Grammar in Section II. Section III describes the characteristics of inflectional languages and in particular of Italian language. Section IV depicts the Morphological Engine and Section VI presents the Smart Grammar Tool. Section VII provides the experimental results to verify our approach. Finally, we draw conclusions and make suggestions for future work in Section VIII.

2. SLU GRAMMAR

A major obstacle to the uptake of the spoken dialog technology is the effort required to realize spoken dialog applications, in particular the CFG grammars to the specific application. These hand-crafted approach can lack robustness when error rates rise or unexpected syntactic forms are used. In contrast, fully statistical approaches to semantic parsing offer the potential of reduced deployment cost, increased robustness, portability and on-line adaptation to improve and extend domain coverage. Realizing this potential, however, is not straightforward [5]. The primary difficulty is that to construct a model with the expressive power of context-free parsing rules such as in the hierarchical Hidden Understanding Model [6] or the hierarchical HMM [7] requires fully-annotated training data and this is expensive to generate. Context-free models provide too many degrees of freedom and in practice, models trained in this way do not seem to converge on useful solutions [8]. The finite-state parsing models such as that used in AT&T's Markov model-based CHRONUS [9] is essentially a HMM-based semantic tagger which although capable of being robustly trained, is not capable of representing hierarchical structure in the data. Alternative the HVS model extends the basic discrete Markov model by expanding each state to encode the stack of a pushdown automaton [5]. Our work does not want to use the statistical approach but continue the aim of facilitating the develop of hand crafted-grammar enhanced domain coverage of a Neo Latin Language.

In this paper we extends our previous research [10,11] in which we addressed the problem of the multi-session management and of a hand-crafted self-learning grammar. We present an alternative implementation of the SLU Grammar of an SDS for inflectional languages. The new version will introduce benefits such as: minor grammar development effort, better grammar representation and structure, easier grammar management and easier user behavior prediction. This is achieved by using a morphological analyzer [12] of a specific inflectional language. We will show a test evaluation upon a case scenario in the Italian language, but the same idea may be extended and reused for other inflectional languages Dialog Systems.

3. PHOENIX PARSER

The Phoenix semantic parser is an easy and robust SLU parser and it is used into the Olympus framework [13]. This parser maps input word strings on to a sequence of semantic frames:

- Named set of slots, where the slots represent related pieces of information.
- Each slot has an associated CFG that specifies word string patterns that match the slot

The CFG describes all the sentences that the system can understand in a meaningful, readable and synthetic form. For SDSs that allow slightly more flexibility to the user, the number of possible sentences is so huge that it becomes impossible to list them all. For example, a user could say "HELLO", "GOOD MORNING" or "HI THERE", but they have the same meaning from the SDS's point of view.

The Phoenix Grammar [14] contains one or more grammar-slots that define a specific meaning by a list of sentences. For example:

```
[greeting]
(hello)
(hi there)
(good morning)
;
```

However, inside a grammar-slot we can describe several inflected forms with the same meaning. In this case, “greeting” is a grammar slot that represents the meaning of the sentences like hello, hi there and so on. For example, Neo Latin Languages have different inflections of words and most of them change the canonical form of the word by adding different suffixes. This means that the parser grammar has to contain all the possible forms of a single word. The software developer has to predict all possible word forms used by the user. A bad prediction will result in a not complete grammar and therefore in sets of sentences not recognized by the parser.

3. INFLECTIONAL AND ITALIAN LANGUAGES

The inflectional or Romance languages are the direct continuation of Latin that has a very rich dictionary. The Latin lexicon had always been in continuous evolution mainly through the addition of suffixes, for example to create diminutive forms. The creation of new forms through the addition of suffix is also a characteristic of the Romance languages. There are many inflectional languages, like Italian, Portuguese, Spanish, French, Provençal and Romanian. A characteristic of the Romance languages is, as in Latin, the creation of inflections. The Romance languages are highly inflectional, in which each inflection does not change the part of speech category, but the grammatical function. In general, the inflected forms are obtained by adding to the root of a canonical form a particular desinence (but there are some irregular cases in which also the root changes, this phenomenon is called apophony).

4. MORPHOLOGICAL ENGINE

The aim of the Morphological Engine is to provide sentences with all words in their canonical form for the Phoenix Parser. For this reason, a morphological engine for the Italian language has been used. In Italian, a word can have different meanings in a sentence: the word “cammino”, for example, can be a verb (in English “to walk”) or a noun (in English “path”). So, it is necessary to choose the right form. For this reason, the morphological engine has been combined with a syntactic analyzer that is a specific version of the Link Grammar Parser (LGP) [15] for the Italian language. It solves cases of ambiguity and it provides sentences containing all the words in their canonical form. In this way, the Phoenix Parser does not have to know all the inflected forms, but only the canonical forms. Following paragraphs describe each component of the Morphological Engine in detail.

4.1. Link Grammar Parser (LGP)

The morphological engine has been combined with a syntactic analyzer that is a version of the LGP for the Italian language. Originally, the LGP has been created for the English language at the Carnegie Mellon University. The LGP is based on the link grammar, a theory of English syntax that builds relations between pairs of words. Each group of words is associated with a set of rules that determine these relationships. In the standard version, the LGP loaded all words in a very large dictionary that for each “tag” included all inflected forms associated. By the fusion with the algorithm of morphological analysis the LGP receives as input a vocabulary with only the words of the phrase. Each rule refers to a “tag” that is associated to the words with particular characteristics. For example, there was a “tag” associated with all transitive verbs in the third singular person, another associated with the feminine singular nouns, one for prepositions, etc. Through the analysis of grammatical categories and the desinences, the morphological engine associates a particular “label” to each word of the sentence. This label contains information regarding the inflection and the grammatical category to which the word belongs. In this way, the LGP receives as input only a string in which each label corresponds only to any words in the sentence. Precisely, each label corresponds to a particular “tag”, each of which is associated with

different syntactic rules. This mechanism allows the LGP to reconstruct more quickly the parse tree of the sentence.

5. SMART GRAMMAR

We focus our attention on the NLU module, which is usually composed by a parser, a grammar associated to the parser and a confidence annotator. The Olympus framework uses the Phoenix parser, by performing a syntactic analysis associating a meaning to user input utterances. The syntactic rules are stored in a context-free grammar. In our previous work [4,Z], we described a dynamic grammar for Olympus by introducing a self-learning mechanism and an intermediate grammar format for a specific grammar generator. In this paper, we address our attention on the management of a context-free grammar for inflectional languages. Our implementation, *Smart Grammar*, reduces the length and the complexity of grammar specification files allowing an easier grammar development and management. It also reduces the prediction effort of the software developer because it makes it possible to recognize all word inflections by only storing its canonical form. Our tool modifies the usual information flow within an SDS introducing a preliminary phase of analysis. Specifically, *Smart Grammar* introduces a morphological analyzer, which modifies each user utterance before it is processed by the parser. A normal utterance flow goes through the Speech Recognition module, the NLU module, the Dialog Manager, the Language Generator module. Smart Grammar consists of a new module inside the Olympus framework. Figure 2 shows the exact collocation of the new morphological analyzer module within the Olympus framework.

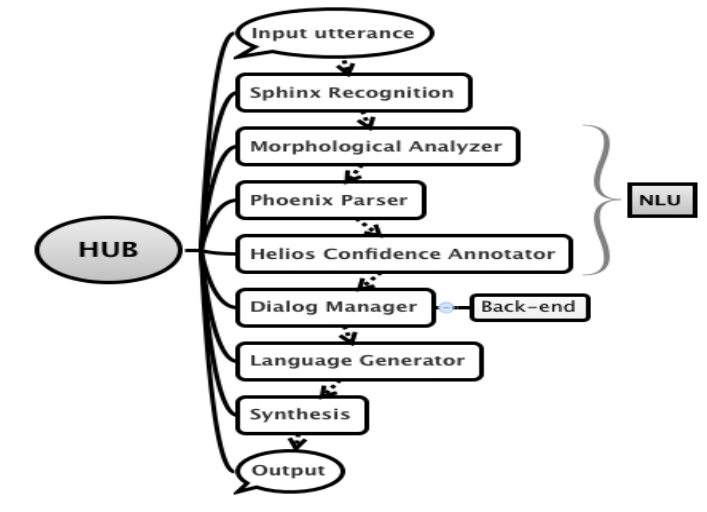


Figure 1 - Algorithm of morphological analysis

In the example, we can see how, by analyzing the structure of the sentence, the morphological analyzer is able to recognize those words and so to transform them in the canonical form. The canonical form is stored in the grammar definition files and therefore the parser will recognize it. This is extremely helpful because it will help reducing the amount of information that have to be stored in the grammar, reducing development time, developing effort and simplifying the grammar structure and management. The idea and results are shown in a simple use case for the Italian language. However, this tool may be easily extended and used in other inflectional languages by using a specific morphological analyzer for each of them.

6. EXPERIMENTAL RESULT

In this section, we will show the benefits of using Smart Grammar to develop an SDS for the Italian Language. Suppose our SDS implements a room reservation system. The programmer must associate the meaning of "I want a room", thinking about all the possible combinations of the user utterances. An example of the Phoenix grammar is described below (the English translation is shown in italics after the arrow "→"):

[need_room]

(([VOLERE] [ARTICOLO] *[GRANDE] [STANZA])
→([TO_WANT][ARTICLE] *[BIG] [ROOM]))

[VOLERE] → [TO_WANT]	[ARTICOLO] → [ARTICLE]	[GRANDE] → [BIG]	[STANZA] → [ROOM]
(voglio) → <i>(I want)</i>	(una) → <i>(a an)</i>	(grande) → <i>(big) (s.)</i>	(stanza) → <i>(room)</i>
(vorrei) → <i>(I would like to)</i>	(alcune) → <i>(some)</i>	(grandissima) → <i>(very big) (f.s.)</i>	(stanzetta) → <i>(small room)</i>
(volevo) → <i>(I wanted)</i>	(poche) → <i>(few)</i>	(supergrande) → <i>(extra big) (s.)</i>	(stanzuccia) → <i>(little room)</i>
(rivoglio) → <i>(I want again)</i>		(supergrandissima) → <i>(very extra big) (f.s.)</i>	(superstanza) → <i>(super room)</i>
....		(grandi) → <i>(big) (p.)</i>	(superstanzetta) → <i>(super small room)</i>
		(grandissime) → <i>(very big) (f.p.)</i>	(stanze) → <i>(rooms)</i>
		(stanzette) → <i>(small rooms)</i>
			(stanzone) → <i>(big room big rooms)</i>
			...

The new grammar format supported by Smart Grammar is:

[need_room]

(volere [ARTICOLO] *grande stanza) → *(to_want [ARTICLE] *big room)*

ARTICOLO → *ARTICLE*

(una) → *a | an*

(alcune) → *some*
;

The morphological engine will match the several forms to the canonical forms that are defined in the grammar-slot. So, the grammar-slot will contain only the canonical form.

There are two cases:

- **Verbs:** whereas in Italian inflected forms are 57, it is possible to recognize them all using only the present infinitive form. If we consider that most of the verbs could have the prefix "ri-", we can get $57 \times 2 = 114$ inflected forms.
- **Nouns and adjectives:** it is not necessary to add all the forms obtained by adding suffixes or prefixes. The morphological analyzer recognizes 8 substantial suffixes and 9 adjectival suffixes and, for both categories, it recognizes 7 different prefixes.

Considering the example shown above, one row (volere [ARTICOLO] *grande stanza) represents a huge number of forms.

[need_room]

(volere [ARTICOLO] *grande stanza)

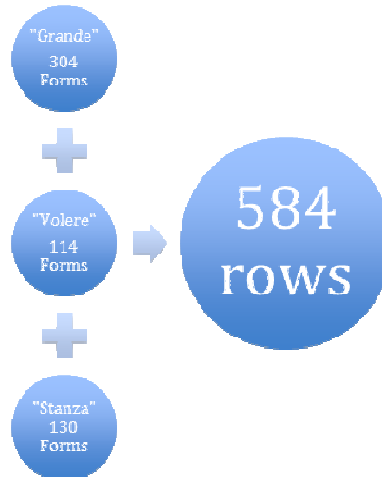


Figure 2 : Example of rows generated

We obtain 114 forms from “volere”.

From “grande” we get the normal inflected forms (in this specific case 2 inflected forms are obtained) plus all the forms derived by adding suffixes. The adjectival suffixes are a total of 9, each is inflected in gender and number, so the algorithm recognizes 36 (4×9) different suffixes. All the obtained forms 38 ($2 + 36$), are added to the forms obtained by the addition of prefixes (7) and we get 266 (7×38) forms. Obviously, some of these are superfluous (specifically the masculine forms and some suffixed forms which really are not used). So to write "grande" corresponds to having 304 forms ($266 + 38$).

From “stanza” we have 2 inflected forms and 8 suffixed forms. In total we get 16 forms (2×8). In addition to these 18 ($2 + 16$) forms the algorithm adds forms with prefixes 112 (7×16). In total there are 130 forms ($112 + 18$).

In conclusion, a line Smart Grammar corresponds to $114 + 304 + 130 = \mathbf{548}$ rows.

7. CONCLUSIONS

In this paper we addressed the problem of developing SDSs with a grammar for inflectional languages. These languages are characterized by words that have different suffixes depending on the conjugations, declensions and alterations. This leads to a high development effort and long grammar specification files hard to structure and to manage. Our implementation *Smart Grammar* introduces a morphological engine, which analyses each user input sentence and transforms every word in its canonical form. Hence, the grammar specification files will contain only the canonical form reducing the amount of information to store. *Smart Grammar* introduces benefits such as minor grammar description file length, a better grammar structure and management, and an easier user behaviour prediction. In our test, using the Olympus framework, we showed how *Smart Grammar* accomplishes a compression rate for grammar description files. So, the Phoenix grammar programmer does not have to worry about matching all possible inflected forms and with suffixes or prefixes. This also prevents the programmer to forget some utterances. Furthermore the size of the Phoenix grammar is reduced even if the SDS is able to recognize a huge number of utterances that the programmer cannot directly know and write in the grammar definition file. Future implementations can reuse the idea of building morphological engines for other inflectional language and use the same principal to help building inflectional SDSs.

ACKNOWLEDGEMENTS

The authors were supported by the Sicilian Region grant PROGETTO POR 4.1.1.1: "Rammar Sistema Cibernetico programmabile d'interfacce a interazione verbale".

REFERENCES

- [1] S. Seneff, "TINA: A natural language system for spoken language applications," Computational Linguistics, vol. 18, no. 1., pp. 61-86, March 1992.
- [2] INFLECTIONAL. Ward, "The phoenix system: Understanding spontaneous Pro-ceedings of ICASSP , 1991.
- [3] Dowd ng, J.; Gawron, J. M. Appel t, D.; Bear, J.; Cherny, L.; Moore, R. and Moran, D. 1993. GEMINI: A natural l anguage system for spokenl anguage understandi ng, 54-61. Col ombus, Ohi o: 31st Annual Meeti ng of the Associ ati on for Computational Linguistics.
- [4] INFLECTIONAL Wang and A Acero, "Evaluation of spoken grammar learning in the atis domain," in Proc Int Conf on Acoustics, Speech, and Signal Processing, Orlando, Florida, 2002.
- [5] Spoken Language Understanding using the Hidden Vector State Model Yulan Hea and Steve Young
- [6] Miller, S., Bates, M., Bobrow, R., Ingria, R., Makhoul, J., Schwartz, R., Jan. 1995. Recent progress in hidden understanding models. In: Proc. of the DARPA Speech and Natural Language Workshop. Morgan Kaufman Publishers, Inc., Austin, TX, pp. 276–280.
- [7] Fine, S., Singer, INFLECTIONAL., Tishby, N., 1998. The hierarchical hidden markov model: Analysis and applications. Machine Learning 32, 41–62.
- [8] Lari, K., Young, S., 1991. The application of stochastic context-free grammars using the inside-outside algorithm. Computer Speech and Language 5 (3), 237–258
- [9] Levin, E., Pieraccini, R., Jan. 1995. CHRONUS, the next generation. In: Proc. of the DARPA Speech and Natural Language Workshop. Morgan Kaufman Publishers, Inc., Austin, TX, pp. 269–271.
- [10] V. Catania, R. Di Natale, A. Intilisano and A. Longo. 2013. A Distributed Multi-Session Dialog Manager with a Dynamic Grammar Parser, International Journal of Information Technology & Computer Science (IJITCS) (ISSN: 2091-1610).
- [11] Vincenzo Catania, Raffalele di Natale, Antonio Rosario Intilisano, Salvatore Biondi, Ylenia Cilano. "An Easy and efficient grammar generator for Understanding Spoken Languages". In press

- [12] Vincenzo Catania, Ylenia Cilano, Raffaele Di Natale, Daniela Panno and Valerio Mirabella, A morphological engine for Italian language, 2nd International Conference on Internet, E-Learning and Educational Technologies (ICIEET 2013), Istanbul - Turkey, 2013.
- [13] Dan Bohus, Antoine Raux, Thomas K. Harris, Maxine Eskenazi, and Alexander I. Rudnicky, "Olympus: an open-source framework for conversational spoken language interface research", in NAACL-HLT '07: Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies, 2007.
- [14] W. Ward, "Understanding spontaneous speech: the Phoenix system," Acoustics, Speech, and Signal Processing, 1991. ICASSP-91, 1991 International Conference on , 14-17 Apr 1991, pp.365-367 vol.1.
- [15] Daniel Sleator and Davy Temperley. 1991. Parsing English with a Link Grammar. Carnegie Mellon University Computer Science technical report CMU-CS-91-196, October 1991.

Authors

Vincenzo Catania received the Laurea degree in electrical engineering from the Università di Catania, Italy, in 1982. Until 1984, he was responsible for testing microprocessor system at STMicroelectronics, Catania, Italy. Since 1985 he has cooperated in research on computer network with the Istituto di Informatica e Telecomunicazioni at the Università di Catania, where he is a Full Professor of Computer Science. His research interests include performance and reliability assessment in parallel and distributed system, VLSI design, low-power design, and fuzzy logic.



Panno received a Dr.Ing degree in Electrical Engineering and a Ph.D. in Telecommunications Engineering from the University of Catania, Italy, in 1989 and 1993, respectively. In 1989 she joined the Department of Informatics and Telecommunications at the University of Catania, where she is now an Associate Professor of Signal Theory. Her research interests are MAN architectures and protocols, traffic management and performance evaluation in broadband networks, fuzzy logic application in the telecommunications field.



Raffaele Di Natale is a Contract Researcher at the DIEEI at the University of Catania. He received his M.Sc. degree in computer science from Catania University, Italy, in 1997 and the Ph.D. in bioinformatics from Catania University in 2012. His research interests include spoken dialog systems, Google Android and data mining.

Email : raffaele.dinatale@dieei.unict.it



Antonio Rosario Intilisano is currently a PhD candidate, University of Catania. His doctoral work explores the Spoken Language Understanding and Automatic Speech Recognition techniques specifically for Neo-Latin Languages.



Ylenia Cilano, is a software engineer at A-Tono. She collaborated with the DIEEI at Univeristy of Catania as contract researcher. Her research interests include spoken dialog systems and morphological engines.

Email: s.saylen@gmail.com.

