

The TextPro tool suite

Emanuele Pianta, Christian Girardi, Roberto Zanoli

Fondazione Bruno Kessler
38100 Trento, Italy
pianta, cgirardi, zanoli{ @fbk.eu }

Abstract

We present TextPro, a suite of modular Natural Language Processing (NLP) tools for analysis of Italian and English texts. The suite has been designed so as to integrate and reuse state of the art NLP components developed by researchers at FBK. The current version of the tool suite provides functions ranging from tokenization to chunking and Named Entity Recognition (NER). The system's architecture is organized as a pipeline of processors wherein each stage accepts data from an initial input or from an output of a previous stage, executes a specific task, and sends the resulting data to the next stage, or to the output of the pipeline. TextPro performed the best on the task of Italian NER and Italian PoS Tagging at EVALITA 2007. When tested on a number of other standard English benchmarks, TextPro confirms that it performs as state of the art system. Distributions for Linux, Solaris and Windows are available, for both research and commercial purposes. A web-service version of the system is under development.

1. Introduction

TextPro is a suite of tools oriented towards a number of NLP tasks such as Web page cleaning, tokenization, sentence splitting, morphological analysis, PoS-tagging, lemmatization, multiword recognition, chunking and NER.

The suite has been designed so as to integrate and reuse state of the art NLP components developed by researchers at FBK. These components were developed under different licenses and sometimes optimized for a single operating system or computer architecture. The user of each component had to manage possible interdependencies with other tools, and solve compatibility or portability issues.

TextPro tries to make it more easy. Single tools are still offered as stand-alone programs, but it is now possible to use them in an integrated environment, providing an extensible framework for creating and adding new components, for both research and commercial purposes. The tool suite has been designed in order to meet the following requirements:

- *Simplicity*: The tool suite should be easy to install, configure and use. A wrapper program allows for specifying what kind of analysis are requested, and takes into account possible interdependencies between tasks.
- *Modularity*: Each tool should have a well defined data interface. Removing, adding or substituting a module should be very easy.
- *Portability*: A tool is taken into consideration for integration in TextPro, only if can be ported to the main operating systems, that is at least Linux, Windows, and Solaris
- *Evaluation*: Whenever possible, tools are evaluated on standard benchmarks.

processors: each processor accepts data from an initial input or from the output of a previous processor, executes a specific task, and sends the resulting data to the next stage, or to the output of the pipeline. Pipelines of processors are widely used in building NLP applications, mainly due to their simplicity and flexibility. GATE is one well-known text processing framework following this approach (Cunningham et al., 2002). Although GATE is a well established and widely used system, it is relatively complex to use. This is also due the choice of using XML stand-off annotation as interchange format between processors. For this reason we decided to implement an alternative framework emphasizing simplicity and easiness of use. This explains also our decision to use tables instead of XML stand-off annotation as interchange format: each module adds its specific information on a different column of the table. The use of the IOB labeling format (Ramshaw & Marcus, 1995) allows the system to annotate a span of tokens with some information (in one column) and another partly overlapping span of words with another kind of information (on a different column). For example:

Spanish	AJO	spanish	B-NP	B-MISC
Farm	NN1	farm	I-NP	O
Minister	NN1	minister	I-NP	O
Loyola	NP0	Loyola	I-NP	B-PER
de	NP0	de	I-NP	I-PER
Palacio	NP0	Palacio	I-NP	I-PER
had	VHD	have	B-VP	O
earlier	AV0	earlier	I-VP	O
accused	VVN	accuse	I-VP	O
Fischler	NP0	fischler	B-NP	B-PER
an	AT0	an	B-NP	O
EU	NN1	eu	I-NP	B-ORG
farm	NN1	farm	I-NP	O
ministers	NN2	minister	I-NP	O
'	PUQ	'	B-NP	O
meeting	NN1	meeting	I-NP	O
.	PUN	.	O	O

The TextPro architecture is based on a pipeline of

The example shows how it is possible to annotate the

span of tokens with different annotations: *Loyola de Palacio* is a Named Entity of type PERSON (PER) and at the same time it is included in the sequence *Spanish Farm Minister Loyola de Palacio* that belongs to a unique chunk of type NP. Each token is also annotated with part-of-speech and lemma. The same purpose is obtained in stand-off annotation by using multiple annotation files pointing to a common hub text.

Of course, having chosen to implement our own framework, gives us also complete control on the framework, and on the licensing policy. This is important in a dynamic research group, whose software development activity is often based on experimental and fast prototyping.

2. TextPro

TextPro consists of nine main components, namely:

- CleanPro: cleaning web pages
- TokenPro: tokenization
- SentencePro: sentence splitting
- MorphoPro: morphological analysis
- TagPro: Part-of-Speech tagging
- ChunkPro: phrase chunking
- EntityPro: Named Entity recognition
- LemmaPro: lemmatization
- MultiwordPro: multiword recognition

The tool suite integrates all these components providing a unique command line interface for users and applications. Almost all components take in input a table with a token on each line and annotations from other components on the columns, and add a new column with the information specific to the tool. The only exceptions are CleanPro, which takes in input an HTML page and returns a text, and TokenPro, which takes in input a text and returns a one column table, with a token in each row.

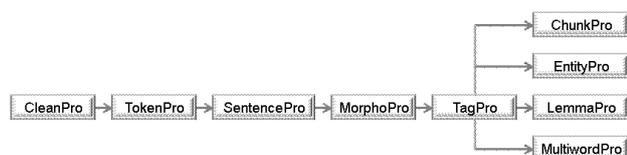


Figure 1: TextPro's architecture

As a general rule, a component is developed under these rules:

- available under LGPL license; the resulting system can be distributed for both research and commercial purposes.
- no programming language constraint; each module can be written in any programming language (i.e. perl, java, c++, etc.).

The TagPro, ChunkPro and EntityPro modules share much of the same architecture. All these components are based on YamCha¹, that is a generic, customizable, and open source text chunker that can be adapted to a number

of other NLP tasks. Using a state-of-the-art machine learning algorithm called Support Vector Machines (SVMs), first introduced by Vapnik (1995), YamCha allows for handling both static and dynamic features, and for defining a number of parameters such as window-size, and algorithm of multi-class problems (pair wise/one vs rest).

In spite of the fact that YamCha is a crucial component of TextPro, it can be easily substituted by any equivalent environment. For example, in the internal version of TextPro, pos-tagging can be performed with both TnT or YamCha, whereas, due to the restrictive TnT license, externally only the YamCha version is distributed.

Concerning performance, TextPro scored as the best system at EVALITA 2007 (Magnini & Cappelli, 2007) for NER and PoS tagging. Tested on CleanEval, a shared task on cleaning arbitrary Web pages, and on CoNLL-2000 (Tjong Kim Sang, 2000) and CoNLL-2003 (Tjong Kim Sang, 2003) shared tasks, TextPro confirms that it performs as state of the art system.

2.1 CleanPro

CleanPro removes mark-up tags and irrelevant text (i.e. words used as navigation menu, common header and footer, etc.) from HTML pages (Girardi, 2007). The resulting text is formatted with a basic encoding of the page structure based on a minimal set of symbols marking the beginning of headers, paragraphs and list elements.

Selection of relevant text in HTML pages is based on the following expectations: the average length of sentences in the relevant section of the text is higher, and the number of links is lower; also, the number of function word in the irrelevant section is lower.

CleanPro is written in Java, and its performance has been evaluated in the CleanEval competition. It produces 62.2% of accuracy on the *Text and Mark-up* task, 80.1% on *Text Only*, 74.0% on the merge of the two tasks (fourth position, 0.7% under the best system).

2.2 TokenPro

TokenPro is a rule based tokenizer that parses the stream of characters of the input text and gives as output a sequence of tokens, each token on a new line. As a general rule, blanks and punctuation marks are taken as token boundaries. However TokenPro recognizes a number of special tokens that can not be handled by the general rule. The source version of TokenPro is a Prolog library, made up of a tokenization engine and a declarative configuration file, regulating tokenization criteria. From the source version a compiled stand alone version is derived. Precision is around 98%.

2.3 SentencePro

SentencePro is a rule based sentence splitter. It takes as input a sequence of tokens and marks the end of the sentence with the special <eos> mark. It also recognizes a number of dot-ending abbreviation. SentencePro exploits knowledge about known abbreviations. A specific abbreviation list identifies a set of dot-ending abbreviations which are not used as end of sentence, even if the following token starts with an uppercase letter. In practice SentencePro can change the tokenization output of TokenPro, beyond marking end of

¹ <http://chasen.org/~taku/software/yamcha/>

sentence.

2.4 MorphoPro

MorphoPro is a morphological analyzer. It is made up of a development environment, implemented in Prolog, and a run-time version implemented in C++. The development version contains a declarative representation of the knowledge needed to analyze and synthesize Italian and English words. This knowledge is used to produce a list of all forms, which is then compiled in a very compact and efficient Finite State Automata. Such FSA is actually used by the run-time version of MorphoPro to analyze English and Italian words. For each input word, MorphoPro produces all possible morphological analysis. A morphological analysis is a sequence of features separated by “+”. The first two features of any analysis are always the lemma and lexical category, followed by a variable list of other features such as gender, number, etc.

2.5 TagPro

Part of speech tagging is the problem of determining the correct parts of speech of a sequence of words. We used YamCha, to build TagPro (Pianta & Zanoli, 2007), a PoS-tagging system able to exploit a rich set of linguistic features, such as prefixes, suffixes, orthographic information (e.g. capitalization, hyphenation), and the morphological features produced by MorphoPro. Each of these features is extracted for the current word, and for the previous and following words. We refer to these features as static features, as opposed to dynamic features, which are decided dynamically during tagging. For the latter, the system uses the tag of the two tokens preceding the current token.

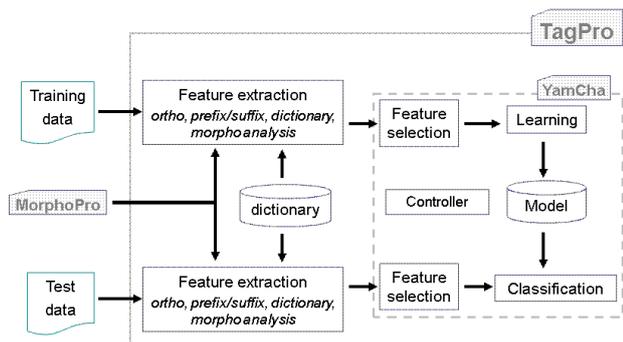


Figure 2: TagPro's architecture

TextPro scored as the best system in the Italian Pos Tagging task at EVALITA 2007, with an accuracy of 98.04%. For English, the system obtains an accuracy of 97.80 when evaluated on the British National Corpus (BNC). Practical annotation time is 1000, 2000 tokens/sec.

2.6 ChunkPro

Text chunking consists of dividing a text in syntactically correlated parts of words and it can be considered as an intermediate step towards full parsing.

For example, the sentence “*He reckons the current account deficit will narrow to only # 1.8 billion in September.*” can be divided as follows: [NP He] [VP reckons] [NP the current account deficit] [VP will

narrow] [PP to] [NP only # 1.8 billion] [PP in] [NP September] .

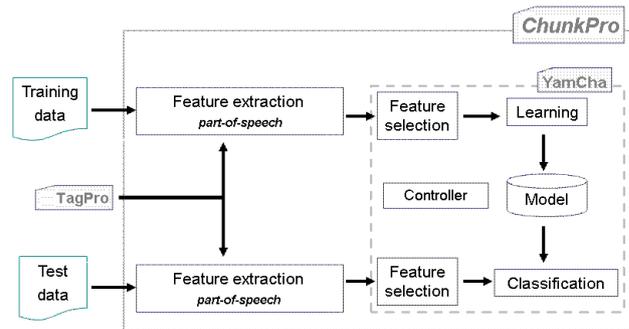


Figure 3: ChunkPro's architecture

Given the word itself and the output of TagPro, ChunkPro groups words into flat constituents of the type: nominal, verbal, adjectival, adverbial. ChunkPro is based on YamCha, which is the system performing the best in the CoNLL-2000 Shared Task with an F1 measure equal to 93.48%.

When tested on the same test corpus, using the part-of-speech annotation given by TagPro, ChunkPro produces an F1 measure of 95.28%. The tool does not work for Italian yet.

2.7 EntityPro

NER is a subtask of Information Extraction which aims at locating and classifying words in text into predefined categories such as persons, organizations, locations, time expressions, etc.

The most frequently applied techniques for this task are based on machine learning: Hidden Markov Models, Maximum Entropy Models, Support Vector Machines (SVMs).

EntityPro (Pianta & Zanoli, 2007) is based on YamCha which exploits SVMs. As argued by T. Joachims (1998), one of the advantages of SVMs is that dimensionality reduction is usually not needed, as they are robust to overfitting and scale up well to high feature dimensions. YamCha allows for handling both static and dynamic features, and for defining a number of parameters such as window-size, parsing-direction (forward/backward) and algorithm of multi-class problems (pair wise/one vs rest).

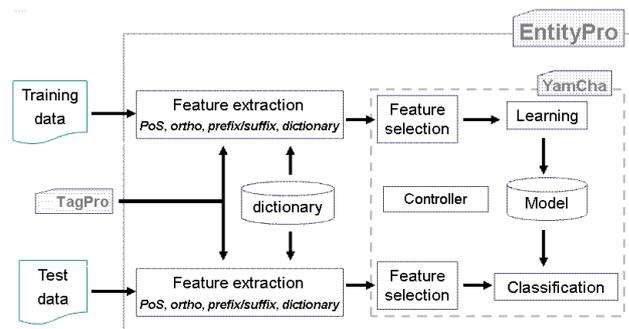


Figure 4: EntityPro's architecture

EntityPro recognizes Italian and English Named Entities, exploiting a rich set of linguistic features, like the PoS produced by TagPro, orthographic information,

collocation bigrams and the occurrence in proper nouns gazetteers. For Italian, these features were extracted from the Italian Content Annotation Bank (I-CAB) developed at FBK.

EntityPro scored as the best system in the Italian NER task, at EVALITA 2007 producing an F1 measure of 82.14%. When tested on CoNLL-2003 for English NER, the tool performs with an F1 of 84.49%. Concerning the speed of annotation, common values are from 1000 to 5000 tokens/sec.

2.8 LemmaPro

LemmaPro is a lemmatizer. Given the morphological analyses produced by MorphoPro and the PoS tag produced by TagPro, it selects all possible lemmas. Although in the vast majority of cases the lemma is unique, in a restricted number of cases LemmaPro can produce more than one lemma.

2.9 MultiWordPro

This tool recognizes occurrences of multiword expression in English and Italian texts, based on a list of multiword specifications providing information about the level of flexibility of each multiword (e.g. token or lemma free or fixed order). Only contiguous multiword are currently handled.

2.10 TextPro Wrapper

The TextPro wrapper allows for specifying what kind of analysis are requested, and takes into account possible interdependencies between tasks. For instance, morphological analysis requires tokenization, and PoS-tagging requires morphological analysis.

Suppose that we need tokenization, sentence splitting, PoS tagging and NER for the following text:

Both Mary and George went to London.

Then, we can give the following command:

```
textpro -l eng -c token+sent+pos+entity <input file>
```

TextPro's output will be:

Both	-	AV0	O
Mary	-	NP0	B-PER
and	-	CJC	O
George	-	NP0	B-PER
went	-	VVD	O
to	-	PRP	O
London	-	NP0	B-LOC
.	<eos>	PUN	O

TextPro provides the requested information using all needed components. We don't need to specify whether morphological analysis is necessary or not. A Web demo version of the system is available at <http://textpro.fbk.eu>.

3.0 Conclusion

We have presented TextPro, a suite of modular tools for analysis of Italian and English texts, developed at FBK. The system tries to combine simplicity, modularity, portability and accuracy.

Plans for the future include the development of an Italian version of ChunkPro, and testing of the Conditional Random Fields machine learning algorithm for PoS tagging, chunking and NER.

Although we find that efficiency of all tools is at least acceptable, we are also aware that there is room for improvements and for better engineering of some of the components. Whenever possible, components were evaluated on standard benchmarks. In some cases, as for MorphoPro, modules were tested in a informal or indirect way; for the near future we are planning to evaluate them on official tasks so as to make results comparable with other systems.

At the same time we plan to analyze in more detail the effects of error propagation through the cascade of modules; we want to investigate how the errors made by each component propagate and degrade the performance of the following modules. Finally we are planning to make the system available as a web-service.

4.0 Acknowledgments

This work has been funded partly by the following projects: the ONTOTEXT sponsored by the Autonomous Province of Trento under the FUP-2004 research program, and partly by the Meaning and PATExpert (<http://www.patexpert.org>) projects sponsored by the European Commission.

We wish to thank Taku Kudo and Yuji Matsumoto for making available YamCha.

5.0 References

- Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V.. GATE: A framework and graphical development environment for robust NLP tools and applications. In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, 2002.
- Girardi, C. (2007). Htmcleaner: Extracting the Relevant Text from the Web Pages. In *Proceedings of WAC3 2007*, 3rd Web as Corpus Workshop, Louvain-la-Neuve, Belgium, September 15-16.
- Magnini, B., Cappelli, A. (2007). EVALITA 2007: Evaluating Natural Language Tools for Italian. In *Intelligenza Artificiale*, Special Issue on NLP Tools for Italian, IV-2.
- Pianta, E., Zanoli, R. (2007). TagPro: A system for Italian PoS tagging based on SVM. In *Intelligenza Artificiale*, Special Issue on NLP Tools for Italian, IV-2.
- Pianta, E., Zanoli, R. (2007). Exploiting SVM for Italian Named Entity Recognition. In *Intelligenza Artificiale*, Special Issue on NLP Tools for Italian, IV-2.
- Ramshaw, L., Marcus, M. (1995). Text Chunking Using Transformation-Based Learning. In Proceedings of the Third ACL Workshop on Very Large Corpora, pages 82-94. Cambridge, MA, USA, 1995.
- Thorsten, J. (1998). Text categorization with support vector machines: learning with many relevant features. *Proc. of ECML-98, 10th European Conference on Machine Learning*.
- Tjong Kim Sang, E. Buchholz, S. (2000). Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal.

- Tjong Kim Sang, E. De Meulder, F. (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003, pp. 142-147.
- Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer.