Portuguese Language Processing Service

Eraldo R. Fernandes†‡∗
efernandes@inf.puc-rio.br
Ruy L. Milidiú†
milidiu@inf.puc-rio.br
Cícero N. dos Santos†
nogueira@inf.puc-rio.br

† Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente 225
Rio de Janeiro, Brasil
‡ Laboratório de Automação, IFG
Rua Riachuelo 2090
Jataí, Brasil

ABSTRACT

Current Natural Language Processing tools provide shallow semantics for textual data. These kind of knowledge could be used in the Semantic Web. In this paper, we describe F-EXT-WS, a Portuguese Language Processing Service that is now available at the Web. The first version of this service provides Part-of-Speech Tagging, Noun Phrase Chunking and Named Entity Recognition. All these tools were built with the Entropy Guided Transformation Learning algorithm, a state-of-the-art Machine Learning algorithm for such tasks. We show the service architecture and interface. We also report on some experiments to evaluate the system’s performance. The service is fast and reliable.

Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing—linguistic processing; H.3.5 [Information Storage and Retrieval]: Online Information Services—Web-based services

General Terms

Algorithms, Performance, Design, Experimentation

Keywords

Portuguese language processing, machine learning, ETL, Web service, RDF

1. INTRODUCTION

There is a strong research effort towards the construction of the Semantic Web. The Semantic Web is based on ontologies [2], generally expressed using the Web Ontology Language (OWL) [13]. The design of these ontologies by Web users is a very hard task [22]. One alternative is to extract semantics from Web textual data using Natural Language Processing (NLP). Zaihrayeu et al. [22] present a NLP system to convert Web hierarchical classifications (directories) to the Semantic Web. A more general NLP system is proposed by Java et al. [9]. Their system is able to extract semantic information from arbitrary textual data and publish it using OWL.

∗Sponsored by a CNPq doctoral fellowship.

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.


1Now accessible at http://agogo.learn.fplf.org.br/fextws.

Supervised Learning could be used to obtain the required Semantic Web metadata from textual information. A critical resource for this Machine Learning approach is the availability of annotated datasets. For textual data, several shallow semantics tasks have already been solved with supervised learning. The simpler tasks help to solve the more complex ones. For each one of them, a corresponding annotated corpus is built [15]. Part-of-Speech Tagging, Text Chunking, Clause Chunking, Named Entity Recognition, Semantic Role Labeling and Dependency Parsing are among the NLP tasks with effective processing systems.

Several NLP tools are already available in the Web [8, 11, 19, 21]. For the Portuguese Language there is just a few. Hence, there is a need for Portuguese NLP tools. There are just a few Portuguese annotated corpora that allow the construction of Supervised Learning based tools.

Milidiú et al. [15] present a set of Portuguese NLP tools. All these tools were built with the Entropy Guided Transformation Learning (ETL) [14], a state-of-the-art Machine Learning algorithm for such tasks.

In this paper, we describe F-EXT-WS1, a Portuguese Language Processing Service that is now available at the Web. The first version of this service provides Part-of-Speech Tagging, Noun Phrase Chunking and Named Entity Recognition. We show the service architecture and interface. We also report on some experiments to evaluate the system’s performance. We observe that the service is fast and reliable. The F-EXT-WS system receives a Portuguese text as input, and attaches the three linguistic information to the text. Besides Semantic Web, these informations are also helpful for tasks such as Semantic Search and Information Extraction.

The remainder of this paper is organized as follows. In Section 2, we briefly describe the Portuguese NLP tasks provided. In Section 3, the ETL algorithm is presented. This algorithm is the F-EXT-WS core engine. The system’s architecture is described in Section 4. The system’s interface is presented in Section 5. In Section 6, we show some experimental results to evaluate the system’s performance and efficiency. Finally, in Section 7, our concluding remarks are presented.

2. PORTUGUESE NLP TASKS

Portuguese tagged corpora is a scarce resource. Therefore, we focus on tasks where there are available corpora. Hence, F-EXT-WS provides the following three Portuguese
Language Processing tasks: Part-of-Speech Tagging (POS), Noun Phrase Chunking (NP) and Named Entity Recognition (NER). In Table 1, we enumerate the three Portuguese corpora used for F-EXT-WS training process. The three tasks are modeled as token classification problems. A token is a word or a punctuation mark and, usually, it is given by a set of features.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Task</th>
<th># Sentences</th>
<th># Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mac-Morpho</td>
<td>POS</td>
<td>53,374</td>
<td>1,221,465</td>
</tr>
<tr>
<td>SNR-CLIC</td>
<td>NP</td>
<td>4,392</td>
<td>104,144</td>
</tr>
<tr>
<td>LearnNEC06</td>
<td>NER</td>
<td>2,100</td>
<td>44,835</td>
</tr>
</tbody>
</table>

These three tasks are interrelated. POS tagging is the most basic among those three, and it is used as input for the NP chunk processor. The NER processor uses POS tags and NP chunks as input features. The remainder of this section describes the three tasks.

2.1 Part-of-Speech Tagging

Part-of-Speech (POS) tagging is the process of labeling each word in a text with a tag that represents its grammar function [10]. POS tags classify words into categories, based on the role they play in the context in which they appear. The POS tag is a key input feature for more advanced NLP tasks. The F-EXT-WS POS Tagger was trained using the Mac-Morpho corpus [1]. In Figure 1, we show a POS Tagged sentence using the column format, which is commonly used in NLP tools. In this format, each line corresponds to a token and each column corresponds to a specific feature. In Figure 1, the first column corresponds to the word feature (the lexical item), and the second column corresponds to the POS tag.

We use the IOB1 tagging style which consists of three tags: O, means that the word is not a NP; I, means that the word is part of a NP and B is used for the leftmost word of a NP beginning immediately after another NP. The column format for the above example is depicted in Figure 2. Observe that, in this example, there is a POS column which contains the POS tag for each token. POS tags are used as an input feature for the NP Chunking processor.

2.2 Noun Phrase Chunking

Noun Phrase Chunking is another basic and important NLP task, consists in break a given sentence in chunks of correlated words (phrases). The most important type of chunk is the noun phrase (NP). A noun phrase is a sequence of words that has the function of a noun in the sentence. In the example that follows, the two noun phrases are enclosed in brackets. The Noun Phrase Chunking task consists in finding all noun phrases in a given text. This task is also approached as a token classification problem by the proposed system.

We approach the NER task as a token classification problem, in the same way as in the CoNLL-2002 shared task [18]. We use a variation of the IOB1 tagging style. The named entities are labeled with I and B tags attached with the entity category. Therefore, we have the tags I-PER, I-LOC, I-ORG, B-PER, B-LOC, etc. The O tag indicates that the word is not a named entity. In Figure 3, we show a column format example for NER. In this example there are two named entities: Brasil and Espanha. The both entities are classified as Location.

2.3 Named Entity Recognition

Named Entity Recognition (NER), among the tasks available in F-EXT-WS, is probably the most important one for Semantic Web applications. NER consists in finding all the proper nouns in a text and to classify them among several given categories of interest or to a default category called Others. Usually, there are three given categories: Person, Organization and Location. The F-EXT-WS NER system classifies named entities among five categories: Person, Organization, Location, Value and Date (time).

We approach the NER task as a token classification problem, in the same way as in the CoNLL-2002 shared task [18]. We use a variation of the IOB1 tagging style. The named entities are labeled with I and B tags attached with the entity category. Therefore, we have the tags I-PER, I-LOC, I-ORG, B-PER, B-LOC, etc. The O tag indicates that the word is not a named entity. In Figure 3, we show a column format example for NER. In this example there are two named entities: Brasil and Espanha. The both entities are classified as Location.

3. ETL ALGORITHM

The F-EXT-WS system engine uses a new machine learning strategy called Entropy Guided Transformation Learning
(ETL) [14]. ETL combines the advantages of two machine learning algorithms: Transformation Based Learning (TBL) [3] and Decision Trees (DT) [20].

TBL is a machine learning algorithm which has been successfully applied to many NLP tasks [3, 6, 5]. This algorithm learns an ordered list of rules that correct classification mistakes in the training corpus. The initial classification is produced by an baseline system, which frequently is just a simple heuristic.

TBL rules are derived from rule templates given as input. A rule template defines a combination of features to be checked when correcting a token classification. Generally, the template set must be created by an application domain expert, and this process is very time-consuming. Indeed, the template generation is the most expensive phase when using TBL and, in the absence of an expert, can lead to bad performance.

Decision Tree learning is one of the most widely used machine learning algorithms. It performs a partitioning of the training corpus using principles of Information Theory. The learning algorithm executes a general to specific search of a training corpus using principles of Information Theory. The main purpose of ETL is to overcome the human driven construction of good template sets, which is a bottleneck on the effective use of the TBL approach. The ETL strategy relies in the use of DT to obtain informative feature combinations. ETL uses a very simple DT decomposition scheme to automatically generate templates. The decomposition process includes a depth-first traversal of the DT. For each visited node, a new template is created by combining its parent node template with the feature used to split the data at that node. The ETL algorithm is depicted in Figure 4.

ETL automatically generates templates using DT learning, next the TBL algorithm is used to generate transformation rules. This process eliminates the need for domain experts, without significant overall performance degradation. In fact, ETL performs better than TBL in most cases. ETL is cheaper and more robust than TBL because it automatically generates effective templates. In addition, in all tested instances, ETL performs significantly better than DT.

4. SYSTEM ARCHITECTURE

The F-EXT-WS design is based on two relevant Software Engineering concepts: Software Product Lines [12, 17] and Software Agents [17]. In this section we describe the main aspects of the system's design and architecture.

The F-EXT-WS design is centered on the task concept. In this context, a task consists of several information pieces: an input text, a linguistic information, an extractor algorithm, an input text, and an output text. Each user must be registered in the system in order to create and execute tasks. The language determines the input text language. There are three persistent entities: User, Task, and FExtFile. These classes represent, respectively, the system users, the submitted tasks, and the input and output files of submitted tasks. They store the registration information about users and tasks. FExtFile objects reference files that store the input and output task data.

Two agents are in charge of scheduling and executing the submitted tasks. The Scheduler agent is a singleton which maintains a queue of ready tasks. This agent determines when a ready task will be polled out from the queue and executed. The Scheduler limits the number of executing tasks in order to control the workload in the server. When a task is scheduled, the Scheduler creates an Executor agent to execute the task. This second agent is in charge of executing the task, carrying the whole process (including execution errors).

Every task in the system has an associated state, whose value is one of the following:

- NEW: new task that was not sent to the Scheduler yet;
- READY: new task that are in the Scheduler queue;
- EXECUTING: executing task;
- CONCLUDED: task whose execution has been completed and its result is available;
- FAILED: task whose execution has been completed, but some problem occurred, and its result is not available.

Figures 6 and 7 depict, respectively, the Scheduler and Executor activity diagrams. As we can see, the Scheduler agent recurrently verifies if some task can be executed. When this agent decides to schedule a task, it creates a new Executor agent. The Executor agent, first, set some properties of the task (execution date and state, for instance). Then, this agent obtains the corresponding Knowledge object from the KnowledgeManager singleton. Next, the Executor agent executes the task dependencies, and, finally, executes the task itself.

The F-EXT-WS design is guided by Software Product Lines (SPL) [12, 17], which is a recent Software Engineering paradigm. SPL aims the development of a family of products to attend a specific domain. A SPL is developed from a common set of core assets, and permits easily derivation of customized products. Very briefly, all the SPL functionalities (features) are divided in two groups: kernel and optional. The kernel features are those present in all products which must be processed by the system. Finally, the output text holds the processing result. We use the column format in the output text. Both the extractor algorithm and the linguistic information options are also extensible. When creating a task, a registered user must provide all these information pieces, except the output file which is generated by the system.

F-EXT-WS uses a scheduler agent to determine the execution order of all submitted tasks and to control the workload in the server. When a task is chosen for execution, the respective extractor is applied for the input text. As soon as the task execution is concluded, its result is stored in an output file that is made available for download.

The F-EXT-WS' classes are divided into three groups: persistent entities, scheduling agents, and extractor engine. The system class diagram is illustrated in Figure 5. There are three persistent entities: User, Task, and FExtFile. These classes represent, respectively, the system users, the submitted tasks, and the input and output files of submitted tasks. They store the registration information about users and tasks. FExtFile objects reference files that store the input and output task data.
The optional features are those present just in specific products. These concepts are used through the whole development process of a SPL such that the derivation of a new product, choosing some optional features, are facilitated. In SPL terminology, a combination of optional features is called configuration.

The F-EXT-WS optional features are the extractor algorithms, the linguistic informations, and the supported input text language. Using the SPL paradigm, we make it possible to easily derive new versions of the service, providing different combinations of NLP tasks.

All the variability in the F-EXT-WS system is managed by a KnowledgeManager object. This is a singleton that is in charge of loading and providing Knowledge objects to the rest of the system. A Knowledge object defines an input language, an extractor algorithm, and a linguistic information. When a user submits a new task, he or she must choose among the Knowledge objects available in the system configuration. New Knowledge objects are easily introduced by some entries in a XML configuration file.

As described in Section 2, some NLP tasks depend on the information provided by another task. For example, F-EXT-WS uses POS tags to perform NP Chunking. The Executor agent is in charge of resolving the task dependencies. As shown in Figure 7, this agent resolves the task dependencies before its execution.

5. SYSTEM INTERFACE

The first version of the system interface is based in HTML forms only. To submit a new task, the user must fill a form and choose an input file from its local file system. The result file is made available through a download link as soon as the task execution is completed. The output file content is formatted using the column format described in Section 2.

Currently, we are working on a Web Service based interface. For this purpose, as the F-EXT-WS is based on the Java programming language, we are using JAX-WS. We also plan to offer RDF output data. We are researching some related approaches that use RDF standard to provide NLP based services. In a short time, we expect to publish a new version of the system interface with this two options.

6. EXPERIMENTAL RESULTS

In this section, we present experimental results that illustrate the F-EXT-WS system performance in two different perspectives. First, we assess the system efficacy for each of the three tasks. Next, we show processing time results.

The ETL strategy achieves state-of-the-art results for the three Portuguese tasks provided by F-EXT-WS. In Table 2, we show, for each task, the performance of the ETL models [15] used in the F-EXT-WS, as well as the performance of other state-of-the-art systems. The POS tagging performance is in terms of accuracy. The performance of the two other tasks are in terms of $F_{\beta=1}$. The $F_{\beta=1}$, which is the harmonic mean between precision and recall, is frequently
Table 2: Performance of ETL and other state-of-the-art systems

<table>
<thead>
<tr>
<th>Task</th>
<th>State-of-the-art systems</th>
<th>Approach</th>
<th>Performance</th>
<th>ETL</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS</td>
<td>TBL</td>
<td>96.60</td>
<td>96.75</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>TBL</td>
<td>87.85</td>
<td>88.61</td>
<td></td>
</tr>
<tr>
<td>NER</td>
<td>SVM</td>
<td>88.11</td>
<td>87.71</td>
<td></td>
</tr>
</tbody>
</table>

As we can see in Table 2, ETL over performs the previous state-of-the-art systems for both POS tagging [4] and NP chunking [3]. For the NER task, ETL is very close to the best reported system, which is based in Support Vector Machines (SVM) [16]. In Table 3, we show the ETL performance in terms of precision and recall for the three tasks provided by F-EXT-WS.

Table 3: ETL performance – precision and recall

<table>
<thead>
<tr>
<th>Task</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS</td>
<td>96.75</td>
<td>-</td>
</tr>
<tr>
<td>NP</td>
<td>88.32</td>
<td>88.90</td>
</tr>
<tr>
<td>NER</td>
<td>86.89</td>
<td>88.54</td>
</tr>
</tbody>
</table>

In order to evaluate the F-EXT-WS system processing time, we carry out experiments using different input texts. The processing time seems to be linear to the size of the input text. In Table 4, we show the average processing times achieved in the experiments for the three tasks. The experiments were performed using the PC that currently hosts the F-EXT-WS service. This PC uses a 1.0GHz Intel
Figure 7: Executor agent activity diagram

Pentium III processor (256KB of cache) and 768 MB of RAM memory.

Table 4: F-EXT-WS computation time per word and per character

<table>
<thead>
<tr>
<th>Task</th>
<th>Words/second</th>
<th>Characters/second</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS</td>
<td>380</td>
<td>2,364</td>
</tr>
<tr>
<td>NP</td>
<td>315</td>
<td>1,951</td>
</tr>
<tr>
<td>NER</td>
<td>248</td>
<td>1,542</td>
</tr>
</tbody>
</table>

7. CONCLUSIONS

In this paper, we describe F-EXT-WS, a Portuguese Language Processing Service that is now available at the Web. The first version of this service provides state-of-the-art Part-of-Speech Tagging, Noun Phrase Chunking and Named Entity Recognition.

The system is designed as a Software Product Line. This architecture is flexible, allowing the smooth introduction of new NLP task processors. We plan to add support for Semantic Role Labeling as soon as a SRL Portuguese Corpus becomes available. We also plan to extend the output format options by providing the RDF standard.

The performed experiments indicate that the service is fast and reliable. In fact, the service can process about 300 words per second, running on a very simple and cheap computer.

We believe that F-EXT-WS contributes to the development of the Semantic Web by providing automatic tools to extract shallow semantics from Portuguese Language textual data.

8. REFERENCES


