Pattern Mining with Natural Language Processing: An exploratory approach

Ana Cristina Mendes[1] and Cláudia Antunes[2]

[1] Spoken Language Systems Laboratory - L^{2}F/INESC-ID
Instituto Superior Técnico, Technical University of Lisbon
R. Alves Redol, 9 - 2°– 1000-029 Lisboa, Portugal
ana.mendes@l2f.inesc-id.pt
[2] Department of Computer Science and Engineering
Instituto Superior Técnico, Technical University of Lisbon
Av. Rovisco Pais 1 – 1049-001 Lisboa, Portugal
claudia.antunes@ist.utl.pt

Abstract. Pattern mining derives from the need of discovering hidden knowledge in very large amounts of data, regardless of the form in which it is presented. When it comes to *Natural Language Processing (NLP)*, it arose along the humans' necessity of being understood by computers. In this paper we present an exploratory approach that aims at bringing together the best of both worlds. Our goal is to discover patterns in linguistically processed texts, through the usage of NLP state-of-the-art tools and traditional pattern mining algorithms.
Articles from a Portuguese newspaper are the input of a series of tests described in this paper. First, they are processed by an *NLP* chain, which performs a deep linguistic analysis of text; afterwards, pattern mining algorithms *Apriori* and *GenPrefixSpan* are used. Results showed the applicability of pattern mining techniques in textual structured data, and also provided several evidences about the structure of the language.

1 Introduction

Recent years have witnessed a great increase of the information available to the main public: information that was formely only accessible on books is now everywhere, and digitally available to everyone. This new paradigm, mainly due to the leverage of the Web, triggered the emergence of several research areas, namely: Information Extraction, Information Retrieval and Question-Answering, in the field of the *Natural Language Processing (NLP)*, and *Text Mining* in the Knowledge Discovery and Data Mining (KDD) field.

Bridging the gap between humans and computers has always been the goal behind *NLP*, by attempting to provide the laters with the ability of understanding and interacting using natural language. Even if this goal was not completely achieved, *yet*, a significant difference can be perceived between techiques that use NLP and other data processing techniques: the first ones have a much deeper understanding of language.

The need for discovering unknown knowledge in text documents set off the interest in *Text Mining*. Also known as Knowledge Discovery in Text (KDT), it aims at discovering and extracting knowledge from textual data, presented in a semi-strutured and unstructured way, and "automatically identifying interesting patterns and relationships" [1]. *Text Mining* techniques have a surface knowledge about the language in which data is presented (for instance, in terms of its stopwords and stemming procedures); usually, they do not possess or deal with any knowledge about the structure of the language.

In this paper we study the application of pattern mining algorithms in textual stuctured data. We aim at going a step further from what it is nowadays accomplished with *Text Mining*. Our hypothesis is that pattern mining over written data will benefit if we take advantage of the deeper understanding of the language that *NLP* provides. Our approach relies on pattern discovery over the processed texts using state-of-the-art *NLP* tools; therefore, in this paper we present the results of a set of studies concerning the usage of typical pattern mining algorithms over text data to which was included linguistic knowledge.

Since *NLP* is the starting point in our pipeline, traditional pre-processing techniques, based on bag-of-words approaches, can not be applied; raw text is used instead. Natural language is ambiguous and profits from an incomparable richness and variety; hence, the usage of *NLP* can imply the introduction of certain errors, namely misclassification of words in morphological classes and wrong division of sentences into its syntatic constituents. Performance, however, should not be considered an obstacle to NLP anymore due to the massification of applications implemented in distributed environments.

The remainder of this paper is organized as follows: Section 2 makes a literature review on this topic; Section 3 formalizes the problem of mining patterns in textual structured data; Section 4 describes the studies we performed on a real-world dataset. The paper finishes in Section 5, where conclusions are drawn and future work directions are presented.

2 Literature Review

Data Mining is an interdisciplinary field that brings together techniques from statistics, database, artificial intelligence and machine learning. *Data Mining* is all about mining patterns in large databases in order to extract hidden knowledge. It gave its first steps in the business domain, when companies experienced an enormous increase of their structured data digitally stored.

A different perspective to pattern mining arose with the outgrowth of large amounts of unstructured text data: *Text Mining*. Although common approaches to *Text Mining* are based on a flat representation of words (usually with little linguistic processing), the close interaction between *Text Mining* and *NLP* is an indesputable fact. They share a strong common ground: text presented in an unstrutured way, written in natural language.

Recent discussion suggests that the integration of linguistic concepts and *NLP* techniques to benefit *Text Mining* applications is still in its fresh start. Not

long ago, in 2005, the usefulness of *NLP* in *Text Mining* was an open issue [2, 3] and an hard question to be answered. The general conclusion arose that more experiments were still necessary. Researchers could not agree on a common opinion about whether *NLP* helps or not the results already achieved by *Text Mining*.

There is a significant amount of approaches to the problem of *Text Mining* stated in the literature that claim the use of results from *NLP*. For instance, *NLP* tags are used in the InFact [4] system for text analysis and search. Linguistic knowledge, namely grammatical and syntactic roles, is used to build *subject-action-object* triples, which are then utilized as indexes. The subsequent search is based on a linguistically enriched query language and allows users to search for actions, entities, relationships and events.

However, and unlike the InFact system, common approaches do not make direct manipulation of *NLP* morpho-syntactic tags to benefit *Text Mining* and search. They rather employ techniques borrowed from Information Extraction (IE) or are related with Text Classification.

In what concerns IE, it regards the extraction of specific data from natural language texts, often by applying *NLP* techniques, like named entities recognition. The usage of IE in *Text Mining* can be seen especially in the field of bioinformatics [5, 6]. Concerning the Text Classification task, it aims at assigning a correct class to previously unseen records, given a collection of already labeled records.

Text Classification is studied, for instance, in [7], in which pattern mining over syntactic trees is used to make an opinion analysis of documents. *Text Mining* techniques for document classification are applied mainly to spam-filtering.

Although the previously cited works only regard the implications of *NLP* in pattern mining, it is worth to mention that the interaction between both is, in fact, bidirectional. The work by [8, 9], for instance, devise methods for named entity recognition and co-reference resolution using association rules.

3 Problem Definition

Natural languages can be formally modelled by Context-Free Grammars (CFG). A CFG is a tuple $G = (N, \Sigma, R, S)$, where: N is a set of non-terminal symbols; Σ is a set of terminal symbols; R is a set of rules, each of the form $A \rightarrow \beta$, in which A is a non-terminal and β is a string of symbols from the set $(\Sigma \cup N)$; and $S \in N$ is the start symbol. A CFG can be thought of in two ways: a generator of sentences or a device that assigns a structure to a sentence [10]. Besides, notice that, in the NLP domain, the set of terminals is related with the lexicon, or vocabulary, constituted by the words and symbols of a given natural language. Also, non-terminals represent either lexical classes of words (Part-Of-

Speech (PoS)[3]) or multi-word constituents of the language (chunks). [4] Figure 1 shows a lexicon and a set rules that belong to a sample grammar G_0[5].

Grammar G_0

$$
\begin{aligned}
S &\rightarrow \text{NP VP} \mid \text{NP VP PP} \\
NP &\rightarrow Pronoun \mid Noun \mid Determiner\, Noun \mid Determiner\, Adjective\, Noun \\
VP &\rightarrow Verb \mid Verb\ \text{NP} \\
PP &\rightarrow Preposition\ \text{NP}
\end{aligned}
$$

Lexicon for G_0

$$
\begin{aligned}
Noun &\rightarrow car \mid man \mid plane \mid morning \mid panter \mid airport \\
Pronoun &\rightarrow I \mid me \mid you \mid it \\
Verb &\rightarrow saw \mid drove \mid want \mid like \\
Adjective &\rightarrow big \mid pink \mid green \mid tall \\
Determiner &\rightarrow the \mid a \mid an \mid this \\
Preposition &\rightarrow on \mid to \mid from
\end{aligned}
$$

Fig. 1. Set of rules for the grammar G_0 and its lexicon.

A *labeled rooted tree* is defined as an acyclic connected graph, denoted as the quintuple $T = (V, E, \Sigma, L, v_0)$ where: V is the set of nodes; E is the set of edges that connect nodes; Σ is the set of labels; $L : V \rightarrow \Sigma$ is the labeling function, that assigns labels from Σ to nodes in V; and v_0 is the root node. The level of a node v is the length of the shortest path from v to the root node v_0. Each node can be either a leaf or an internal node: leafs have no children and internal nodes have one or more child nodes. An ordered labeled rooted tree is a labeled rooted tree where the children of each internal node are ordered.

A natural language sentence can be generated by a sequence of rule expansions, commonly represented by a *parse tree*. Thus, a *parse tree* is defined as an *ordered labeled rooted tree* τ, in which $v_0 = S$ and tree nodes are labeled according to the morphological and syntactic classes of each corresponding unit. Figure 2 shows the tree structure of the sentence "I saw a pink plane on the airport" according to G_0.

Mining patterns using algorithms which do not focus on this specific data structure (like sequential pattern mining algorithms) implies a transformation of parse trees into other structures with different characteristics. In this context, the *l-sequence* of an ordered tree is the preorder transversal label sequence of all nodes; the preorder transversal label-level sequence, *l^2-sequence*, is the cannon-

[3] Traditional grammar classifies words according to eight PoS, or lexical classes: verb, noun, pronoun, adjective, adverb, preposition, conjunction, and interjection.

[4] For the sake of simplicity, we will refer to the terminal and non-terminal symbols of a given sentence as lexical units and morpho-syntactic units, respectively.

[5] Symbols NP (noun phrase), VP (verb phrase) and PP (prepositional phrase) are language constituents (chunks) with a specific syntactic function within a sentence. Readers interested on more information about this topic are suggested to refer to [10].

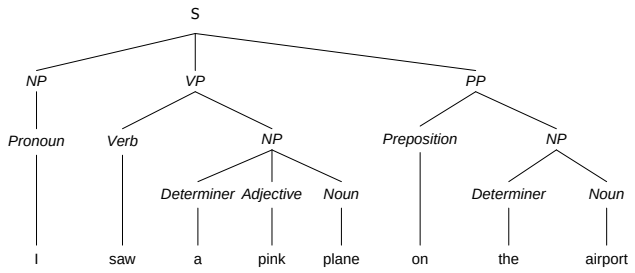

Fig. 2. Parse tree of "I saw a pink plane on the airport" according to G_0.

ical representation of a tree, composed not only by the label of each node, but also by its level in the tree. Both concepts are introduced in [11], where authors also prove the uniqueness of l^2*-sequences*.

In the *l-sequence* of a parse tree each item has a linguistic meaning and is related with a morpho-syntactic or semantic property of a determined unit within a sentence. Likewise, the l^2*-sequence* of a parse tree provides the knowledge and a unique representation of its structure. Therefore, transforming sentences into *l-sequences* (or l^2*-sequences*) represents the transformation of the unstructured dataset into its corresponding structured version.

Given an input sentence S, the preorder transversal sequence of its associated parse tree is a new sentence S' provided with a deeper linguistic knowledge, which captures the morpho-syntactic structure of S. We define the sentence S' as a *textual structured sentence* since its base constituents refer to natural language units, either lexical (words) or morpho-syntactic (PoS and/or chunks).

Given a database D of transactions and a user-specified minimum support σ, a pattern is said to be frequent if it is contained in, at least, σ transactions in D.

Problem statement. The application of pattern mining algorithms over textual structured data provides deeper knowledge about a natural language inner organization than tradicional *Text Mining* techniques. Given a text database and a user-specified minimum support, the problem of mining textual structured data is to discover the frequent structures of a natural language.

Being so, this exploratory study attempts to bring the understanding of the natural language structure that *NLP* provides to pattern mining. We propose to test typical algorithms on textual structured data in order to discover patterns on structured data. The high-level view of the proposed approach is shown in Figure 3.

4 Case Study

Experiments described in this section were conducted on a dataset composed by news articles from a Portuguese daily newspaper, *Público*, dated from years 1994 and 1995. Since the news domain is far too broad, only the news belonging

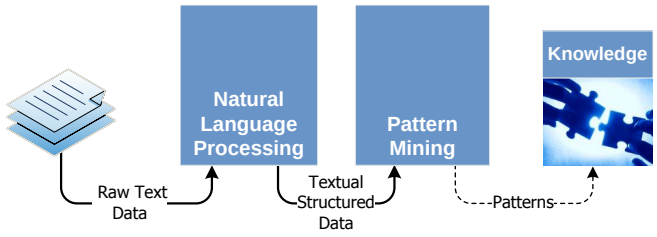

Fig. 3. Our exploratory approach to discovering patterns in text.

to the category "*National News*" were used. Moreover, from these, paragraphs with less than 100 characters were discarded, since these often refer to articles' title. The characteristics[6] of the dataset are shown in Table 1.

Table 1. Characteristics of the dataset.

National News

Size (MB)	Words (#)	Distinct Words (#)	Paragraphs (#)	Sentences (#)
$\approx$ 27	$\approx$ 3 000 000	85 770	58 744	176 447

Text data was processed using the Spoken Language Systems Laboratory (L^2F) *NLP* chain for Portuguese, running on a computacional grid [12]. The chain involves the following processing steps: morphological analysis (by Palavroso [13]), morphological disambiguation (by MARv [14]), multi-word contraction and term spliting (by RuDriCo [15]) and, finally, chunking, named entities recognition and dependencies extraction (by XIP [16]). The chain is responsible for creating the parse tree, morpho-syntactically disambiguated, for each sentence received as input. An illustrative example of the output of the *NLP* chain for the input sentence "O primeiro-ministro José Sócrates inicia uma visita oficial à Madeira."[7] is depicted in Figure 4.

Thus, the parse tree used in our experiments is defined as τ, and nodes in τ are labeled according to a function that marks a node v with a *PoS* or a *chunk* if v is an internal node, or with a *language word* or a *language symbol* if v is a leaf. In addition, *PoS* belong to the set of symbols: {`NOUN`, `VERB`, `ART`, `ADJ`, `ADV`, `CONJ`, `PRON`, `INTERJ`, `PREP`, `PUNCT`, `PASTPART`, `REL`, `FOREIGN`, `SYMBOL`, `NUM`}; a *chunk* is one of {`TOP`, `VP`, `NP`, `ADVP`, `AP`, `PP`, `SC`}.

[6] Notice that the *Paragraphs* column concerns a rough text spliting by carriage return, while the *Sentences* column is related with the correct natural language sentences identified by the morpho-syntactic analyser. Thus, one paragraph might contain several sentences.

[7] "Prime-minister José Sócrates starts official visit to Madeira." For clarity reasons, this sentence will be referred to and used as reference sentence throughout the rest of this paper.

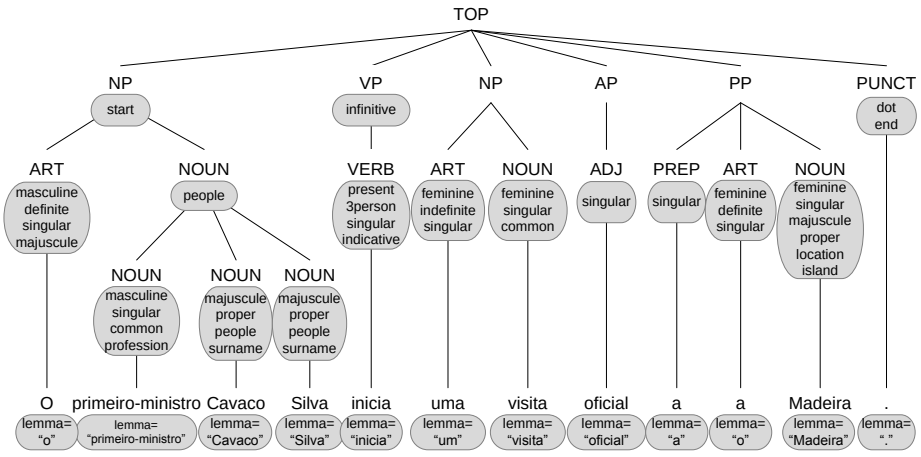

Fig. 4. Parse tree of “O primeiro-ministro José Sócrates inicia uma visita oficial à Madeira.”

Considering the reference sentence, its associated textual structured sentence is the sequence: TOP NP ART O NOUN NOUN primeiro-ministro NOUN José NOUN Sócrates VP VERB inicia NP ART uma NOUN visita AP ADJ oficial PP PREP a ART a NOUN Madeira PUNCT .

Since the *NLP* chain introduce many features in each tree node, associated with the morphological, syntactic and semantic analysis (some of which are shown in small rounded boxes in Figure 4), those unwanted are discarded by a post-processing stage. This post-processing aims at building the alphabet for the mining algorithms with only the relevant symbols for our experiments. If all features were used, a much larger alphabet would be created. However, this would lever the discovery of many frequent noisy patterns, not serving the purpose of our current research studies.

4.1 Pattern Mining Algorithms over Textual Structured Data

There are several algorithms to mine patterns in data. They can follow specific categorizations depending on the characteristics of the input data: items in a basket, sequences, tree structures, and so forth. In these studies, however, we slightly transformed our dataset to be used as input for two pattern mining algorithms, which belong to different strands on KDD.

The first algorithm employed in the experiments was *Apriori* [17]. *Apriori* is based on a candidate generation and test philosophy, following the anti-monotone property, according to which if a pattern is not frequent, then none of its supersets is frequent. In this case, each textual structured sentence is seen as a different transaction in the database and each of it components is considered is a different item.

The second algorithm used was *GenPrefixSpan* [18]. This algorithm for sequential pattern mining represents a generalization of *PrefixSpan* [19] to deal with gap constraints, while maintaining a pattern-growth philosophy: avoids the candidate generation step altogether, and focus the search on a specific set of the initial database. Here, and like with *Apriori*, the textual structured sentences are transactions and its components are items. However, in this case, the position of each component is related with a discrete time instant; being so, each component occurs immediately after its left side neighbour. This algorithm was used to mine sequences of contiguous components in textual structured sentences.

A sample input for each pattern mining algorithm is presented in Table 2. Each line contains a textual structured sentence, configured for the correspondent algorithm, as previously defined in Section 4. Since we will present results from tests over different input data, we will refer to this input configuration as the *default configuration*.

Table 2. Input data for each pattern mining algorithm (default configuration).

Algorithm	Input
Apriori	TOP NP ART O NOUN NOUN primeiro-ministro NOUN, José NOUN Sócrates VP VERB inicia ...
GenPrefixSpan	TOP(1) NP(2) ART(3) O(4) NOUN(5) NOUN(6) primeiro-ministro(7) NOUN(8) José(9) NOUN(10) Sócrates(11) VP(12) VERB(13) inicia(14) ...

4.2 Results

The results of each pattern mining algorithm, *Apriori* and *GenPrefixSpan*, are displayed on Table 3 and Table 4, respectively. A detailed analysis of the results allow us to draw further conclusions about the language. It is worth to remind, however, that these experiments were conducted on a very distinctive dataset, composed by news articles focused on a particular domain: the "*National News*". These are commonly written in a well-formed language, with reliable and formal discourse structures and few spelling errors. Tests on other datasets with different characteristics would probably lead to different results. One may argue that, regardless the text's characteristics, it is still Portuguese we are dealing with, and its structure remains unchangeable; however, some conclusions we draw shall not be extended to the language as a whole, as it is spoken and written in other real-world situations.

Regarding the results from the *Apriori* algorithm, the great increase of patterns discovered as the values of minimum support dicrease should be pinpointed and deserves a careful analysis. Actually, since language PoS and chunks are not in great amount (≈ 20), some of which populating every or almost every sentence in the textual structured data (consider, for instance, constituents `TOP`, the top-level chunk that refers to a sentence, and `PUNCT`, the PoS that refers

Table 3. Results of the *Apriori* algorithm.

Support (Transactions) Min (%)	Max (%)	Discovered Patterns (#)
75	100	1 227
50	100	41 079
25	100	1 454 135
25	50	1 413 056
10	25	28 721 028

to punctuation), it is not a surprise to verify that a significant part of the frequent patterns are composed only by morpho-syntactic units and stopwords. This, however, does not seem to bring any reliable knowlegde about language's structure. Also, most of these patterns have no linguistic meaning or sense at all, since they lack on order.

Table 4. Results of the *GenPrefixSpan* algorithm (default configuration).

Min Support (Transactions) %	#	Discovered Sequences (#) Total	Size > 1
50	88 224	42	20
25	44 112	95	66
10	17 645	235	197
5	8 822	401	346
2.5	4 411	790	698
1	1 764	2 139	1 885
0.5	882	4 962	4 405
0.25	441	11 875	10 715
0.1	176	36 375	33 807

In what concerns the results from the *GenPrefixSpan* algorithm, the number of discovered sequences is much smaller when compared to the number of discovered patterns discovered with *Apriori*, given the same support. Indeed, the philosophy behind *GenPrefixSpan* avoids the generation of patterns with no linguistic meaning. This algorithm mined only subsequences of our textual structure sentences; therefore with linguistic meaning.

Also, it is worth to mention that in 50% of our dataset only 22 one-unit-size sequences were discovered. From these 22, only 6 were language symbols or language words, namely: comma, dot, determiners "a" and "o" (English word "the"), preposition "em" (corresponding to English words "in", "on" and "at") and preposition "de" (English word "of"). All chunks are frequent given a mininum support of 50% and, having in mind that the *NLP* chain labels nodes according to 15 different PoS, the following lexical classes were not considered as being frequent: `INTERJ`, interjections, `PARTPART`, words in past participle, `REL`,

relative pronouns, FOREIGN, words written in other language than Portuguese, SYMBOL, special symbols, and NUM, numerals.

A sample output of both algorithms is shown in Table 5.

Table 5. Patterns discovered by each pattern mining algorithm (default configuration).

Algorithm	Pattern	Support (%)
Apriori	SC ADVP a de PP . PUNCT	25.8
	CONJ AP a PRON PP . NP TOP	31.6
	AP ADJ a PREP VP VERB NOUN PUNCT TOP	50.1
	o . PREP VP VERB	65.4
GenPrefixSpan	NP NOUN NOUN durão NOUN barroso	0.5
	VP VERB era NP	1.0
	PREP em NOUN lisboa	1.1
	que VP VERB	22.8

Besides the default configuration, similar other experiments were conducted on the same dataset with *GenPefixSpan*. These, however, differ on the amount of encoded linguistic knowledge, in order to discover how mined patterns evolve given different domain conditions. The tested configurations are listed bellow:

(I) The textual structured sentences contain the level of each unit; the cannonical representation of each tree was used: TOP-L0 NP-L1 ART-L2 O-L3 NOUN-L2 NOUN-L3 primeiro-ministro-L4 NOUN-L3 José-L4 NOUN-L3 Sócrates-L4 VP-L1 VERB-L2 inicia-L3 NP-L1 ART-L2 uma-L3 NOUN-L2 visita-L3 AP-L1 ADJ-L2 oficial-L3 PP-L1 PREP-L2 a-L3 ART-L2 a-L3 NOUN-L2 Madeira-L3 PUNCT-L1 .-L2

(II) The same as the default configuration, however *PoS* components are removed. Therefore, each textual structured sentence is composed only by language words and symbols and chunks: TOP NP O primeiro-ministro José Sócrates VP inicia NP uma visita AP oficial PP a a Madeira .

(III) The same as the default configuration, however only language words and symbols are used. Being so, instead of textual structured data, only the linguistically processed text is used as input: O primeiro-ministro José Sócrates inicia uma visita oficial a a Madeira .

(IV) Only typical *Text Mining* pre-processing techniques are applied: stopwords and punctuation removal. The dataset is not linguistically processed; therefore, no linguistic knowledge is included at all. Moreover, albeit the dataset was the same, the input for the pattern mining algorithm is slighlty different: instead of each transaction having a direct correspondence with a correct natural language sentence, an entire paragraph is now related with a transaction, since the *NLP* chain is not used: primeiro-ministro José Sócrates inicia visita oficial Madeira encontra-se amanhã presidente governo regional

The two graphs in Figure 5 compare the results of applying the algorithm *GenPrefixSpan* with different input configurations, either in terms of total frequent sequences and in terms of frequent n-unit-size sequences ($n > 1$).

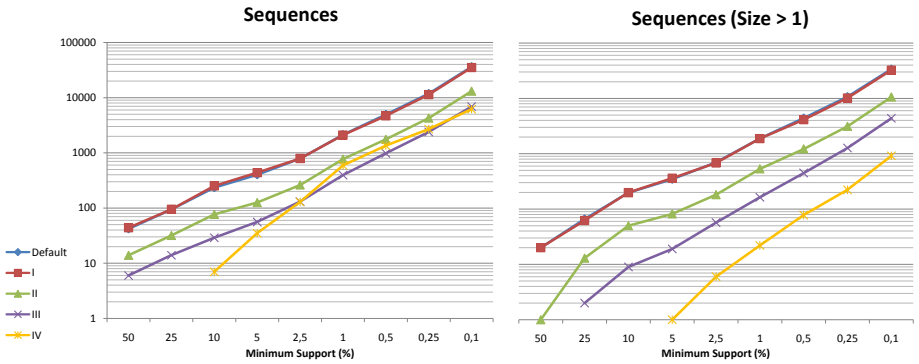

Fig. 5. Number of discovered frequent sequences given different input configurations to the *GenPrefixSpan* algorithm.

As it can be perceived from the analysis of both graphs, no significant differences exist between the number of discovered patterns using configurations **Default** and **I**. However, the cardinality of the alphabets used in the two configurations is fairly distinct: 85 792 and 116 634 symbols, respectively. This can bring us to the conclusion that the structure of the language in our dataset is somewhat static, and frequent nodes do not often change their level in the parse tree. A careful analysis of the results showed that the discovered sequences are indeed almost the same in both configurations. Differences are mainly due to two main reasons: for a determined minimum support value x, one sequence in the **default** configuration is associated with several sequences in configuration **I**, and their support is higher that x; for a determined support value x, one sequence in the default configuration is associated with more than one sequence in configuration **I**, but their support is lower that x. For instance, `[NP PRON] (support 47.0%)` is a sequence in the **default** configuration associated with sequences `[NP-L1 PRON-L2] (39.7%)` and `[NP-L2 PRON-L3] (13.0%)` in configuration **I**. For a minimum support of 10% the later sequence is frequent; this does not occur, however, for a minimum support of 25%.

The comparison of one-unit-size frequent sequences between configurations **default** and **I** provides some clues about the level of frequent words, PoS tags and chunks in the parse tree. Ultimately, and although this is not this work's main purpose, these results can be used to uncover potential errors in the textual structured data, introduced by the morpho-syntactic analyser: for instance, when it comes to conjunctions, for a minimum support of 0.1%, they are frequent one-unit-size sequences in more than 60% of the transactions (`[CONJ] (61.2%)`); if we consider also the levels of nodes, conjunctions are frequent when placed in

levels 1, 2, 3 and 4 of the parse tree (`[CONJ-L1] (54.9%)`, `[CONJ-L2] (15.0%)`, `[CONJ-L3] (1.0%)` and `[CONJ-L4] (0.2%)`, respectively). Given these results, one might wonder whether the last pattern is a misclassification of words in their morphological class or an error in the sintactic division of sentences, introduced by the NLP chain when processing the raw text dataset.

Regarding the other configurations, greater differences exist among results. In every configuration in which the NLP chain was applied, the number of discovered sequences got smaller with the decrease of linguistic knowledge in the input data. For instance, with a minimum support of 2.5%, 790 sequences were discovered in the **default** configuration, 263 in **II** and 131 in **III**.

Configuration **II** allow us to understand how sentences are composed regarding its bigger substructures:

- in terms of those substructures inner organization, for instance:
  - a Noun Phrase usually starts with determiners "o" and "a"
    ⇒ Frequent patterns: `[NP o] (support 45.2%)` and `[NP a] (36.1%)`.
  - the sequence of words "o presidente de a república" ("the president of the republic") is a frequent Noun Phrase
    ⇒ Frequent pattern: `[NP o presidente de a república] (0.4%)`
- in terms of those substructures interaction with other sentence constituents, for instance:
  - in almost 40% of the cases, a sentence begins with a Noun Phrase
    ⇒ Frequent pattern: `[TOP NP] (38.8%)`.
  - it is more likely that a Noun Phrase appears after the inflected form of the verb *to be* "é", than a Verb Phrase
    ⇒ Frequent patterns: `[é NP] (6.0%)` and `[é VP] (2.0%)`.

Although a strict comparison can not be made between configuration **IV** and the others (after all, transactions are different), results show similar numbers in terms of discovered sequences for configurations **III** and **IV**; however, the number of sequences with size bigger than one is much lower when the stopwords and the punctuation were removed. Indeed, the usage of sentences without any pre-processing technique, led to the appearance of many n-unit-size sequences composed uniquely by those small and frequent words, with no content or semantic value (stopwords) and by punctuation.

4.3 Critical Analysis

The application of pattern mining algorithms over textual structured data brought us diverse clues about the natural language inner organization: similar results could not be achieved if only a flat representation of words were used.

Also, recall that the algorithms were applied on the output of an already mature and stettled *NLP* chain. No other knowledge was manual or automatically inserted on our test data, besides some small modifications we made in order to test the various input configurations.

Moreover, these studies prove the fact that some algorithms better apply in this specific domain than others: *Apriori* led to a combinatorial explosion of

results, since it does not keep track of the input's order; on the contrary, *GenPrefixSpan* seemed to discover reliable frequent language substructures. However, the discretization of parse trees into sentences and subsequent application of a sequential pattern mining algorithm implies the discovery of incomplete substructures (like, for instance, the sequence `PP PREP a ART`). In fact, other mining algorithms should be further explored in this specific domain, given the output of state-of-the-art linguistic analysers.

5 Conclusions and Future Work

In this paper we presented a series of exploratory studies we conducted in order to discover patterns on text data. The notion of textual structured data was introduced as being a database of text sequences which contain specific knowlegde about a sentence's morpho-syntactic structure.

We took advantage of the knowledge of the language that *NLP* provides and used it as input for two different pattern mining algorithms: *Apriori* and *GenPrefixSpan*. On one hand, *Apriori* discovered a large amount of frequent patterns; however, these had little meaning since the order between the units within the natural language sentences was not preserved. On the other hand, *GenPrefixSpan* output several sequences that led to a better understanding of the natural language, Portuguese in this case.

As future work we intend to explore the syntactic organization of the language and employ pattern mining algorithms over tree structures. We would like to test how these algorithms apply to the NLP domain. In addition, we aim at investigating the impact of including other morphological features (for example, words' gender, number and lemma) in the mining process. Also, our results are confined to the output of an NLP chain, that includes itself the understanding of the language in the form of a grammar (for Portuguese); therefore, the results reflect the structure of that grammar. Since the application of pattern mining algorithms to textual structured data is a language independent approach, we would like to test it on other languages, with distinct morpho-syntactic structures.

References

1. Feldman, R., Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press (2006)
2. Kao, A., Poteet, S.: Report on KDD Conference 2004 Panel Discussion Can Natural Language Processing Help Text Mining? SIGKDD Exp Newsl **6**(2) (2004) 132–133
3. Kao, A., Poteet, S.: Text mining and natural language processing: introduction for the special issue. SIGKDD Explor. Newsl. **7**(1) (2005) 1–2
4. Liang, J., Koperski, K., Nguyen, T., Marchisio, G.: Extracting Statistical Data Frames from Text. SIGKDD Explor. Newsl. **7**(1) (2005) 67–75
5. Leser, U., Hakenberg, J.: What Makes a Gene Name? Named Entity Recognition in the Biomedical Literature. Briefings in Bioinformatics **6**(4) (2005) 357–369

6. Otasek, D., Brown, K., Jurisica, I.: Confirming protein-protein interactions by text mining. In: SIAM Conference on Text Mining. (2006)
7. Matsumoto, S., Takamura, H., Okumura, M.: Sentiment Classification Using Word Sub-sequences and Dependency Sub-trees. In Ho, T.B., Cheung, D., Li, H., eds.: Proceedings of PAKDD'05, the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining. Volume 3518 of Lecture Notes in Computer Science., Hanoi, VN, Springer-Verlag (2005) 301–310
8. Budi, I., Bressan, S.: Association rules mining for name entity recognition. In: WISE '03: Proceedings of the Fourth International Conference on Web Information Systems Engineering, Washington, DC, USA, IEEE Computer Society (2003) 325
9. Budi, I., Bressan, S., Nasrullah: Co-reference resolution for the indonesian language using association rules. In Kotsis, G., Taniar, D., Pardede, E., Ibrahim, I.K., eds.: iiWAS. Volume 214., Austrian Computer Society (2006) 117–126
10. Jurafsky, D., Martin, J.H.: 12. In: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall (2008)
11. Wang, C., Hong, M., Pei, J., Zhou, H., Wang, W., Shi, B.: Efficient pattern-growth methods for frequent tree pattern mining. In Springer, ed.: Advances in Knowledge Discovery and Data Mining: Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining. (2004)
12. Luís, T.: Paralelização de Algoritmos de Processamento de Língua Natural em Ambientes Distribuídos. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal (2008)
13. Medeiros, J.C.: Análise morfológica e correcção ortográfica do português. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal (1995) (in Portuguese).
14. Rodrigues, D.J.: Uma evolução no sistema ShRep: optimizacão, interface gráfica e integracão de mais duas ferramentas". Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal (2007) (in Portuguese).
15. Paulo, J.: Extracção Semi-Automática de Termos. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal (2001) (In Portuguese).
16. Aït-Mokhtar, S., Chanod, J.P., Roux, C.: A multi-input dependency parser. In: IWPT, Tsinghua University Press (2001)
17. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc. (1994) 487–499
18. Antunes, C.M.: Pattern Mining over Nominal Event Sequences using Constraint Relaxations. PhD thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal (2005)
19. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: ICDE '01: Proceedings of the 17th International Conference on Data Engineering, Washington, DC, USA, IEEE Computer Society (2001) 215–226