Use of Co-occurrences for Temporal Expressions Annotation

Olga Craveiro[1,2], Joaquim Macedo[3], and Henrique Madeira[2]

[1] School of Technology and Management, Polytechnic Institute of Leiria, Portugal
[2] CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
{marine,henrique}@dei.uc.pt
[3] Department of Informatics, University of Minho, Portugal
macedo@di.uminho.pt

Abstract. The annotation or extraction of temporal information from text documents is becoming increasingly important in many natural language processing applications such as text summarization, information retrieval, question answering, etc.. This paper presents an original method for easy recognition of temporal expressions in text documents. The method creates semantically classified temporal patterns, using word co-occurrences obtained from training corpora and a pre-defined seed keywords set, derived from the used language temporal references. A participation on a Portuguese named entity evaluation contest showed promising effectiveness and efficiency results. This approach can be adapted to recognize other type of expressions or languages, within other contexts, by defining the suitable word sets and training corpora.

1 Introduction

The Web is actually a key information source for our daily lives. Search engines are essential to use efficiently the information available at the Web. Therefore, there is an intensive academic and industrial research effort to improve the efficiency and effectiveness of underlying Web Information Retrieval (IR) models.

Temporal information is a key piece on most information system applications and, consequently, in Web based applications. Nevertheless, it has not been the focus of a systematic and deep work on IR applications. The temporal dimension is an important element of the user's information need context, and if used effectively it would improve the relevance of documents response set. The most effective temporal entities recognition programs on free (or semi-structured) text are heavily dependent on the natural language used in those texts. The typical approaches are based on intensive hand-crafted rules. Another set of solutions are based on natural language independent stochastic models which assigns probabilities to strings in a given language L allowed by the use of a training corpus. Between these two approaches there are a variety of mixed ones. The stochastic approaches are best suitable (and simpler) for multilingual context such as the Web. Additionally, another important requirement for huge Web applications is the efficiency, which can be achieved by improving simplicity of the used models.

In most applications, unigrams, bigrams or trigrams are used due to its simplicity and because they are hard to beat by more complex n-grams. However, the nature of

J. Karlgren, J. Tarhio, and H. Hyyrö (Eds.): SPIRE 2009, LNCS 5721, pp. 156–164, 2009.

natural languages is such that many words combinations are infrequent and can even do not appear in a given training corpus. This point to the need for smoothing techniques to overcome zero probability strings on maximum likelihood estimation.

This work proposes a method for annotating temporal information to be included in a temporal aware Web IR model. The experimental scenario used is a Portuguese text collection but we believe that the proposed approach can be easily adapted to other languages. This method uses simple probabilistic based techniques to recognize temporal entities. Temporal entities are detected using temporal expression patterns derived from the higher probability temporal reference word co-occurrence from training corpora. Less frequent and unseen expressions are ignored. The main advantage of the proposed approach is the simplicity and efficiency improving, preserving a promising effectiveness.

The structure of the paper is as follows: section 2 presents the related work on temporal entities recognition, section 3 details the proposed model, section 4 discusses the results obtained from experimental evaluation and section 5 concludes the paper.

2 Related Work

Although a plethora of works exists for the area of temporal references extraction in English texts, to the best of our knowledge, none of them creates automatically a set of expression patterns and applies it for temporal entities recognition. Expression patterns matching require sentence-by-sentence processing. However, Natural Language Processing systems are mainly based on term-by-term processing, using term linguistic characteristics for its identification, such as techniques presented in [1]. An annotation scheme to represent dates and time, based on a variety of hand-crafted and machine-discovered rules, was proposed in [2]. This approach uses finite-state automata, a common technique in this area. A very different approach was proposed in [3]: the temporal expressions identification in French documents is based on a context-scanning strategy (CSS).

Unlike the English language, Portuguese text language extraction area has not been much explored. In particular, temporal information has not been the focus of any systematic work reported in the literature. PALAVRAS, for instance, is an automatic grammar and lexicon-based parser for unrestricted Portuguese text [4]. This system is an important tool for Portuguese text annotation, even though using a generic approach to handle temporal expressions. More recently, a temporal processor called XTM (XIP Temporal Module) was developed by Hagège and Tannier [5, 6] supporting Portuguese language processing, among others languages, such as, English and French. XTM is rule-based, relying on a word-by-word processing.

The novelty of our proposal relies on having lexical patterns automatically generated from Portuguese texts and follows an inductive empirical approach which starts from the data to the knowledge, unlike the work reviewed above.

3 Annotating Temporal Information

In this section, we present our approach for the recognition of temporal entities. Despite other possible applications for entities recognizing in other contexts and languages, Portuguese language is the focus of our experiments.

The method relies on a set of temporal patterns, based on regular expressions, used to identify and classify the temporal expressions found in Portuguese texts. The patterns are created using words co-occurrence, determined from a set of seed keywords, which are Portuguese temporal references. Our method is based on a two-stage approach, each stage being carried out by a different module: the first stage is executed by the *Co-occurrence processor* module (henceforth COP) and the second stage is carried out by the *Annotator* module. The modules work as follows. Firstly, the COP module creates the temporal patterns, based on the training corpora and on the set of reference words which are divided in two sets: lexical markers and grammatical markers. Then, these patterns are used by the Annotator module to perform the annotation of the Portuguese temporal expressions. Fig. 1 shows a diagram of the model architecture and module interconnection.

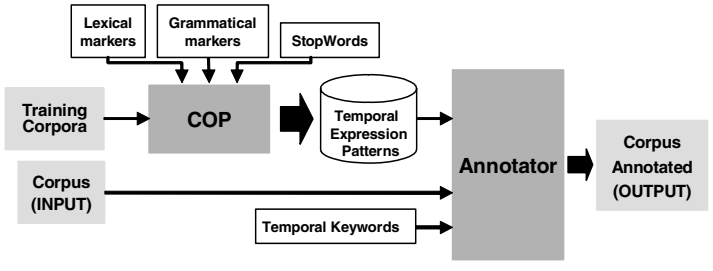

Fig. 1. Model architecture and module interconnection

3.1 Annotation Scheme

The temporal expressions are annotated accordingly as the temporal guidelines defined by the organization and the participants of the Second HAREM[1] [7]. The classification of temporal expressions defined in these guidelines was supported by the annotation scheme TimeML [8]. The annotation comprises a unique identification, a category which is TIME, a type (*calendar_ref*, *duration* or *frequency*) and a subtype only for the type CALENDAR_REF (*date*, *time* or *interval*). A detailed specification of the annotation scheme can be found in [7]. Some examples are presented below. The first sentence exemplifies a date expression and the second sentence represents a temporal expression which expresses a repetition in the time.

```
(1) I was in Berlin <EM ID="1" CATEG="TIME" TYPE="CALENDAR_REF"
    SUBTYPE="DATE">in 2008</EM>.
(2) I visit my parents<EM ID="2" CATEG="TIME"
    TYPE="FREQUENCY">every day</EM>.
```

3.2 Co-occurrence Processor

The task of the COP module is to create a set of temporal patterns that will be used by the Annotator module. COP can be easily executed over various corpora, yielding a

[1] Second evaluation contest of Named Entities Recognizer system, in Portuguese language document collections.

considerable number of patterns that enrich the annotation stage. It is worth noting that the COP module is only needed to get the set of patterns and once the patterns are established the COP module is not used anymore. Nevertheless, the patterns can be fine tuned later on to improve the identification of temporal expressions.

The COP module analyzes the input training corpora, determines the words combination and its frequency, and uses a statistical approach to decide which patterns must be created according to the co-occurrences found. COP module has several execution steps. The first step creates a list composed by the temporal expressions found and their frequency. These expressions were found using the lexical markers. These markers must be composed by all Portuguese words from which temporal expressions can be composed (e.g. months, seasons, weekdays, units of temporal measure like *day*, *week*, *month*, *year*, …). This set of words is used to detect their co-occurrences which are present in a maximum of *n* words before and/or *n* words after. An example of temporal expressions using the Portuguese temporal word *ano* (*year*) and *n=2*: *"No **ano** passado"* (*In the last year*), *"No próximo **ano** de 2010"* (*In the next year 2010*).

In the second step the list of expressions is pruned. Specifically, the expressions which do not make semantically sense in a language context are removed from the list, using the grammatical markers. However, the expressions that just contain lexical markers and grammatical markers are kept in the list, as long as no stopword exists in neighborhood. For example, in the sentence *"A rua 1 de Maio"* (*The 1st May street*) the expression *"1st May"* is not a temporal expression because it is the name of a street. As the word "*street*" is a stopword, it is excluded.

The next step aggregates temporal expressions found in the previous step according to the following rules. First, the temporal expressions are aggregated if they contain a date or time references. For example, *"Em Abril"* (*in April*) and *"Em Maio"* (*in May*) are aggregated in a single expression with a special tag *"Em tag_MONTH"* (*In tag_MONTH*). Second and last one, the temporal expressions are aggregated if they contain more than one co-occurrence with the same temporal word at the same position. For example, the expressions *"No **ano** passado"* (*In the last year*) and *"No **ano** seguinte"* (*In the following year*) are aggregated in *"No **ano** passado | seguinte"* (*In the last | following year*). The frequency of the aggregated expressions is the sum of the frequency of each expression. The resulting list is ordered by frequency (greater to less). Some expressions can be excluded by a previously defined minimum frequency threshold.

Finally, the patterns are defined by regular expressions. For each pattern is associated the classification according to the temporal guidelines of the Second HAREM (see section 3.1).

3.3 Annotator

The objective of the Annotator module is to identify and classify Portuguese temporal expressions with the relative annotation written in the original text, through the patterns defined by COP. After the text split into sentences, each of one is processed in five steps. The first step is introduced to improve performance by excluding all the sentences that cannot have a temporal expression. Only sentences with date and time references and/or temporal words from Portuguese language defined in a keyword list (see Fig. 1) are processed. For example, the sentence *'Lisbon is the capital of Portugal'* is not processed. However, the sentence *'**Today** is sunshine'* is processed.

The generation of candidate temporal expressions is done in second step. First, it identifies time expressions and date expressions which can be complete or incomplete dates. Then, these expressions are tagged with a "special tag" such as *tag_DATE*, *tag_MONTH*¸ *tag_YEAR*, *tag_WEEK*.

In the third step, the method verifies if the sentences match any temporal pattern. In this case, each sentence is annotated with semantic classification corresponding to the matching pattern (fourth step). Finally, the "special tags" are replaced by the original text.

4 Evaluation

Knowing there is a huge amount of documents to process in common application scenarios, one of the key decisions is to achieve the best tradeoff between efficiency and effectiveness in temporal expressions identification. As our option is to favor efficiency to some extent, with the used configuration the system may not find all temporal expressions (even for a trained human reader, it would be difficult to identify all the temporal expressions, as the notion of time is often subtly embedded in the text).

Our primary goal was to evaluate the performance of the method in a restricted environment. Therefore, the COP was configured only to create patterns of simple temporal expressions, expressions composed by only one temporal word or one date or time, and a maximum of *n* words before and/or *n* words after (*n=2*). The lexical markers were restricted to: months, seasons, weekdays, holidays (*Natal* (Christmas), *Páscoa* (Easter) and *Carnaval* (Carnival)) and the following words[2]: *década, século, ano, mês, semana, dia, hora, minuto, ontem, anteontem, amanhã, hoje, manhã, noite, tarde*. Furthermore, were included in the temporal patterns a set of limited grammatical markers[3] composed by prepositions {*à(s), de, em, durante, desde, pelas, no, naquele, (n)este, (n)esse*}, ordinal adjectives {*anterior, seguinte, próximo, passado, último*} and *haver* (to have) verb conjugations. Note that in the pruning step, the stopwords were not considered yet.

Using the prototype implementation for our method, we have carried out a set of experiments and participated in the evaluation contest Second HAREM with a promising effectiveness and efficiency for the first results obtained (72% precision and 53% recall).

The experiments were performed in a Personal Computer with 1GB RAM memory and an Intel Core 2 E6600 2.4GHz processor, running with Microsoft Windows XP Professional version 2002 SP 2.

We divided the experiments in two tasks: identification and classification. In the identification task, the goal was to obtain complete temporal expressions, while in the classification task the idea was to assign the type and subtype specification. In order to clarify this, we show below some examples with mistakes, accordingly as the temporal guidelines (see section 3.1). The expression *'1909-1955'* is correctly identified.

[2] English version: decade, century, year, month, week, day, hour, minute, yesterday, the day before, tomorrow, today, morning, night, afternoon.

[3] English version: prepositions {in, the, during, for, since, by, (in) this, (in) that}, ordinal adjectives {previous, following, next, past, last}.

However, the classification is wrong, as the subtype must be INTERVAL. The expression *'2009'* is incomplete. The correct identification must be *'in 2009'*, but the classification is right.

```
(1) CATEG="TIME" TYPE="CALENDAR_REF"
    SUBTYPE="DATE">1909-1955</EM>
(2) in CATEG="TIME" TYPE="CALENDAR_REF" SUBTYPE="DATE">2009</EM>
```

For efficiency purposes, we measure the time spent on the Annotator module to identify and classify the temporal expressions in test collection. For effectiveness, we calculate the three usual metrics: precision, recall, and the harmonic mean F (F-measure), using the evaluated collection. The formula used to calculate the classification was defined in [9].

4.1 The Collections

The Second HAREM Collection (2ndHC) was the corpus used in our experiments which texts are structured in different genres, such as journalistic, blog, FAQ, literary, etc., and are written in two Portuguese variants: Portuguese from Portugal and Portuguese from Brazil. The 2ndHC is the test collection that is composed by 1040 documents with 33,712 sentences and 668,817 words. The evaluation test is the Time Gold Collection (TGC), a subset of 2ndHC (30 documents, 622 sentences and 12,992 words) and their documents were manually annotated following time HAREM guidelines [7]. The training collection (TC) was another subset of 2ndHC which is composed by all documents of 2ndHC that do not belong to the TGC. The 2ndHC and TGC collections are available through Linguateca[4] and properly detailed in [9].

4.2 Results

The result of TC processing by COP was 289 patterns which can detect more than 289 different temporal expressions because some of them have more than one combination. Note also that about 17% of these patterns permit the identification of dates and times in different formats.

The execution time of the Annotator module was calculated in two scenarios. Scenario 1 – skipped the first step of the Annotator, but all the sentences are processed by every other steps of this module; Scenario 2 – all steps are executed, therefore, only sentences which we believe could indicate the presence of a temporal reference are processed (see section 3.3). In scenario 2, only 17,525 of 33,712 sentences (52%) proceed to the next step, the processing finishes here to the other sentences and the execution time decreases about 27.5% justified by the missing pattern matching step with the remaining sentences. This way, the performance was improved and the Annotator module processed the test collection with an output rate of about 22KB per second.

The effectiveness results are presented in Table 1. We can observe that the results of the two tasks obtained by our system do not have significant differences, which means that the Annotator module shows the same behavior in the two tasks. So, we

[4] Available at HAREM site http://www.linguateca.pt/HAREM

Table 1. Annotation results: our system versus XIP-L2F/Xerox system

	Identification Task		Classification Task	
	(1) *Our system*	(2) *XIP-L2F/ Xerox system*	(1) *Our system*	(2) *XIP-L2F/ Xerox system*
Precision	84,27%	75,31%	83,05%	73,76%
Recall	64,10%	77,59%	64,23%	75,80%
F-measure	72,82%	76,43%	72,44%	74,77%

can conclude that if this module identifies a given temporal expression, then it will achieve a good success in its subsequent classification. Table 1 also shows the results obtained by the XIP-L2F/Xerox system using the same collections. This system was ranked in the first place in the Second HAREM and its results are presented in [6]. Our approach matches the results of the top system concerning precision, but it shows lower recall. This is mainly due to the restricted set of lexical and grammatical markers used by COP to generate the patterns, which affects recall. However, we believe that we can improve recall by increasing the restricted set used by the COP module. We plan to exploit this in future work.

Although, the COP was configured with $n=2$, which means that the expressions was limited to 5 words, only approximately 12% of TGC expressions have more than two words before and/or after the lexical marker (see Table 2). Furthermore, the $n>4$ only exists about 1% of these expressions. It is our intention to carefully study the variation of the n value, since increasing this parameter makes the COP module more complex.

Table 2. TGC temporal expressions

	# words between temporal word and the expression begin/ending					
	n=2	n=3	n=4	n=5	n=6	n=7
# temporal expressions	205	18	8	1	1	0

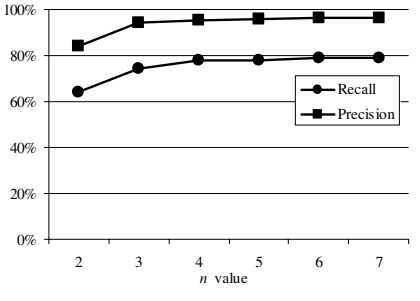

Fig. 2. Precision and Recall values with $2 \leq n \leq 7$

In the precision calculation, the temporal expressions partially correct are not considered. Our system found 178 temporal expressions of which 150 are correctly identified. Indeed, the incorrect expressions are only 3 of 28; the others are incomplete because one or more words are missing in the annotated expression. We analyze the precision variation with n ranging from 3 to 7 (see Fig. 2). We observe that the precision improves with n, namely when n goes from 2 to 3. Improvement is still seen from with $n>3$, but at a lower rate. This means that having COP creating temporal patterns with $n>2$ and one more temporal word improves precision and recall. However, the recall achieved is about 80%. As we said above, the improvement of this metric can be done by increasing the restricted set of markers used by the COP module.

5 Conclusions

The main contribution of this paper is an original method for temporal named entities recognition. The approach creates semantically classified temporal patterns, based on regular expressions, using word co-occurrences obtained from training corpora and a pre-defined seed keywords set, derived from temporal references. The prototype implementation of the proposed method is composed by two modules and some configuration files including temporal reference words and a set of temporal keywords (only used to improve efficiency).

As this temporal named entities recognizer is intended for use in huge Web IR applications, the need for a careful tradeoff between effectiveness and efficiency is the justification for the deliberate simplification of the used approach. Even with a set of limitations and simplifications of a prototype implementation, our method has shown promising results in identification and classification of temporal named entities.

As further work, the most obvious research direction is the variation of used parameters: n (number of maximum words on the temporal expression) and low frequency threshold. Additionally, we plan to tune the lexical and grammatical markers and to improve the pruning step resorting to stopwords to lower the rate of false positives. The method can be also evaluated with foreign languages (English, for instance) and another application contexts (other kind of named entities recognition). To do this, a previous study of the chosen language and context, based on a careful statistical analysis, is needed to define the lexical and grammatical markers.

References

1. Mani, I.: Recent developments in temporal information extraction. In: RANLP 2003, Borovets, Bulgaria, pp. 45–60 (2004)
2. Mani, I., Wilson, G.: Robust temporal processing of news. In: ACL 2000: 38th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, p. 69–76 (2000)
3. Vazov, N.: A system for extraction of temporal expressions from French texts based on syntactic and semantic constraints. In: ACL 2001 workshop on temporal and spatial information processing, Toulouse, France (2001)

4. Bick, E.: The Parsing System, PALAVRAS: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Ph.D. thesis, Dept. of Linguistics, University of Aarhus, Denmark (2000)
5. Hagège, C., Tannier, X.: XTM: A Robust Temporal Text Processor. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 231–240. Springer, Heidelberg (2008)
6. Hagège, C., Baptista, J., Mamede, N.: Reconhecimento de entidades mencionadas com o XIP: Uma colaboração entre a Xerox e o L2F do INESC-ID Lisboa. In: Mota, C., Santos, D. (eds.) Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM. Linguateca (2008)
7. Hagège, C., Baptista, J., Mamede, N.: Apêndice B: Proposta de anotação e normalização de expressões temporais da categoria TEMPO para o HAREM II. In: Mota, C., Santos, D. (eds.) Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM. Linguateca (2008)
8. Pustejovsky, J., Ingria, B., Sauri, R., Castano, J., Littman, J., Gaizauskas, R., Setzer, A., Katz, G., Mani, I.: The Specification Language TimeML. In: Mani, I., Pustejovsky, J., Gaizauskas, R. (eds.) The Language of Time: A Reader. Oxford University Press, Oxford (2005)
9. Mota, C., Santos, D. (eds.): Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM. Linguateca (2008)