Identifying, Matching and Sorting Events

Viviana Isabel Guerreiro Grave Cabrita

IST – Instituto Superior Técnico
L^{2}F – Laboratório de Sistemas de Língua Falada – INESC ID Lisboa
Rua Alves Redol 9, 1000-029 Lisboa, Portugal

Abstract. Operations analysis, extraction and processing of text stands out within the Natural Language Processing (NLP) task and are essential for the development and improvement of systems capable of, among others, perform summarizations and translations of text without human intervention. The work described in this paper focused on the identification, matching and temporal ordering of events.
It contributed to the development of the processing chain STRING (Statistical and Rule-Based Natural Language Processing), developed by the Laboratory for Spoken Language Systems of the Institute of Systems Engineering and Computers Research and Development in Lisbon (INESC-ID), as a new module with the aim of matching and sorting events to each other in a timeline.

Keywords Temporal Expressions, Portuguese, Natural Language Processing (NLP), Event Sorting

1 Introduction

STRING (*Statistical and Rule-Based Natural Language Processing*) [1] is a chain of natural language processing, developed for the Portuguese language by L^2F, with a modular structure, rule and machine learning based. It could already parse the text, detecting the structure, extracting a lot of features and filtering a lot of events and temporal expressions. However, it didn't support event ordering. Because of that, the work described in this papper involved the study of portuguese language and the study of the system itself in order to develop and integrate a solution from scratch.

As we'll see, the system is already modular, making it easier to add new functionalities but the event ordering problem is quite complex and extensive to be fully solved in a single step. So, firstly, it was necessary to define which problems would be solved and which conditions should be taken into account.

1. It was defined that there would be only 2 types of event ordering (*before* and *simultaneously*). That was the minimal strictly necessary. Every missing or unknown order would be saw as an *unknown* type, lacking a visual representation.

2. The module would sort only events in the same sentence, as the first approach was to take one step at a time.
3. The module should be scalable, allowing it to evolve in order to achieve bigger goals in the future. The module should allow event ordering between phrases and more distinct types of ordering in the future, as *include* or *intercepted-by*, once wanted or seen to be necessary. It also should not strict the expansion of the system itself. It needed to fit the system design.
4. It was also encouraged a visual representation of the solution, so it would be easier to be interpreted by the human eye.

Knowing this, the next sections describe the problem faced, the analysis made about it, the representation, the design adopted and the evaluation process.

2 Defining the problem

As described in several dictionaries, in a generic way, an *event* [2] is a spatial and temporally locatable occurrence, even if, sometimes, it is not possible to obtain that kind of information. The event definition, being very general, can still be adapted to context in which they are used. Some systems studied, as it'll be seen, had their own definitions, defined as it seamed fit. So, it was necessary to define this concept in this system too.

An event is defined, in this context, as either predicative names (e.g. *entrevista* (*interview*)), verbs (e.g. *entrevistar* (*to interview*)) or other expressions with a predicative value. So, an event can be an measure (e.g. *pesar* (*to weight*)), an psychological state (e.g. *irritar* (*to irritate*)) or an causative predicate (e.g. *causar* (*to cause*)), among other things.

Aquele livro pesa 2 quilos. (*That book weights 2 kilograms.*)
A Joana irritou-se com o João. (*Joana was angry with João.*)
O sismo causou o tsunami. (*The earthquake caused a tsunami.*)

There can be a event chain instead a single event, where one event can be caused by another. The next sentence can be take as an example of these chains, showing an event *adormecer* (*to sleep*)caused by another *estar cansado*(*be tire*). Because of this, he can derive that evento *adormecer* happened after *cansado.*

Ele adormeceu porque estava cansado. (*He felt asleep because he was tired.*)

There are other ways to sort events: by looking at temporal expressions and their relation with events.

O João nasceu a 3 de Março de 1987. A 25 de Abril de 1975, deu-se a revolução dos cravos. (*João was born in 3 March 1987. In 25 April 1975, there was a carnation's revolution.*)

So, as seen, it's possible to sort events in a sentence. And it was intended to solve this sorting problem by search events and temporal relations. And many of those relations can be found by searching connectors, as conjunctions (e.g. *e* (*and*)), prepositions (e.g. *em* (*at*)) and adverbs (e.g. *por conseguinte* (*therefore*)).

Estava mau tempo e decidimos ficar em casa. (*There was a bad weather and we decide to stay at home.*)
We decided to study at night. (*We decided to study at night.*)
Estava mau tempo esta manhã, por conseguinte, decidimos ficar em casa. (*There was a bad weather this morning, therefore, we decide to stay at home.*)

3 State of the Art

There are a few natural language systems which already match and sorts events in other languages, like the system made by Marşic (2011) [3], processing English's texts, or like TERSEO (*Temporal Expression Resolution System Applied to Event Ordering*) (2003) [4], processing Spanish content.

From the analisys of the systems studied, it can be found different models and algorithms applied in order. There are two big paths that can be followed while projecting the solution: the system can be made rule-based or machine learning based. STRING [1], the system expanded with this work is mostly rule-based, which means it uses human knowledge of the language to solve the problems while other systems uses algorithms and models to derive the solution by itself, thought learning and, sometimes, using external sources like lexical databases, WordNet [5–7], or relational databases, VerbOcean [8]. Marşic's system is an example of a advanced system using both rule-based and machine learning techniques and external sources in order to get an perfected outcome.

Mostly of the studied event sorting systems, TRIPS & TRIOS (2010) [9], XTM(*XIP Temporal Module*) (2007) [10] and Marşic system, used an common annotation schema, TimeML [11–14], or one derived from it. As studied it is a well advanced schema which considers not only the sorting order *before* and *simultaneous* but all of Allen's relations [15]. However, STRING as it's own schemas, extended as a result of this work to best fit our needs: only two sorting options for now and adapted to our own features.

Some systems studied, Chambers et al. (2008) [16], NCSU-INDI & NCSU-JOINT [17], XTM and Marşic, divided the sorting problem in two: local, sort events within the same sentence, or global, between different sentences. This method allowed them to focus two smaller problems instead of a big one. NCSU-INDI & NCSU-JOINT, for example, started the solution with the local problem and extended it latter to the global problem.

With the system evaluation made in terms of *precision*, *recall* and *f-measure* metrics, those systems achieved values between 42-88%, 42-88% and 42-84%.

4 Identifying and Sorting Events in Portuguese

There are some issues with event identification in portuguese. Some words, like *aliança* have double meaning: it can mean an *alliance*, seen as an event, or *wedding ring*, an object. Semantic desambiguation is an complex problem which couldn't not be solved at the moment. Many events, however, could be easily identified from predicative verbs and nouns associated with support verbs.

O Pedro já *jantou*. (Peter already *ate*.)
O *jogo* foi animado. (The *game* was thrilling.)

On the other hand, conjunctions, prepositions and some adverbs connects events within the same sentence, giving different meanings according its own functions. As an example, the conjunction *porque* (*because*) give us an cause-effect situation:

O Rui não saiu de casa *porque* precisava de estudar. (Rui didn't left home *because* he needed to study.)

In addition to these, temporal expressions can be directly associated with events or throught connectors, adding extra temporal information to it.

Ele leu o jornal *ontem*. (He read the newspaper *yesterday*.)

After connecting events within each other, the sorting problem can be solved looking at all the data given by both events and connectors involved: some verbs have usefull information based on it's tense, aspect and mode; prepositions and conjunctions express different orders depending on their roles in the sentence.

5 Identifying, Relating and Sorting Events

With the goal of keeping the modular structure of STRING system, it was created a module which encapsulated the solution to add, figure 1, divided into two sub modules one responsible with the extraction of event's relations and other with the goal to use that knowledge to sort those events. This way, not only was expanded the identification of events, but it was introduced a new functionality without changing much.

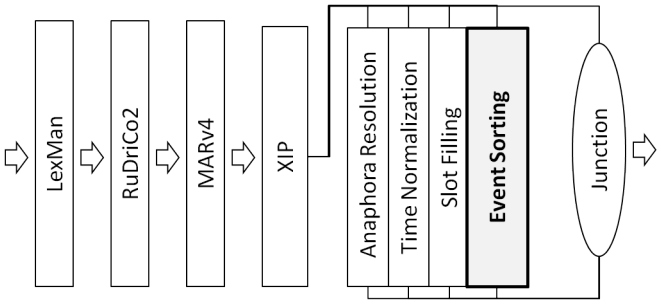

Fig. 1. Representation of STRING with the Event Sorting module.

The first submodule reads the previous identified events, temporal expressions and connectors, extracting and classifying all relations regarding those

events. The second submodule interprets those relations, verifying the temporal and aspetual features associated with it. From every pair of events related with each other, the submodule will try to sort them, creating an order graph as a result.

It was defined 4 types of relations to extract in the first submodule: two of them would define a relation of subordinative/subordinate event with (1) or without a connector (2); another one would define a relation between coordenated events (3); and, the last one, would define a temporal relationship (4).

1. `EVENT_INDIRECTRELATION(≪event≫, ≪event≫, ≪connector≫)`
2. `EVENT_DIRECTRELATION(≪event≫, ≪event≫)`
3. `EVENT_GROUPRELATION(≪event≫, ≪event≫ [,≪connector≫])`
4. `EVENT_TIMERELATION(≪event≫, ≪temporal expression≫)`

The second submodule sort events considering the following conditions: one event ocorrs before another only if it's end ocorrs before the seconds starts; one event is simultaneous with the second if boths happen at the same time or if one of them intersects the other one.

With the dependencies from the previous task and information regarding events, temporal expressions and connectors, the second submodule extracts one of these new types of dependency:

```
EVENT_ORDERBEFORE(≪previous event≫, ≪afterwards event≫)
EVENT_ORDERSIMULT(≪primary event≫, ≪secundary event≫)
```

After extracting all dependencies, every sorting dependency is converted into two nodes (representing each event) and an line between them defining the type of ordering.

As an example, consider the next sentence:

A Maria *acordou hoje antes de* a irmã se *levantar enquanto* ainda *bocejava*.
(Mary *woke up today before* her sister *getting up while* still *yawning*).

During the text processing, among other things, it is identified all events, *acordou* (*woke up*), *levantar* (*getting up*) and *bocejava* (*yawning*), connectors, *antes de* (*before*) and *enquanto* (*while*), and temporal expressions *hoje* (*today*).

Looking for event relations, it is extracted and classified two pairs of events and one temporal relation:

```
EVENT_INDIRECTRELATION(acordou, levantar, antes de)
EVENT_INDIRECTRELATION(levantar, bocejava, enquanto)
EVENT_TIMERELATION(acordou, hoje)
```

From these relations, the module extracts the ordering and creates an order graph from it, represented in the figure 2:

```
EVENT_ORDERBEFORE(acordou, levantar)
EVENT_ORDERSIMULT(levantar, bocejava)
```

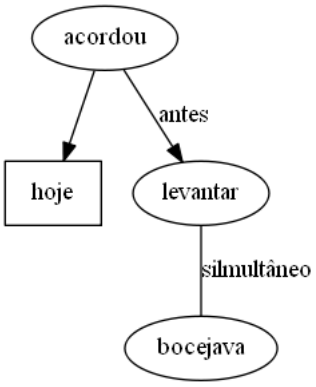

Fig. 2. Graph representating the result of event sorting module.

6 Evaluation

For evaluation proposes, it was build a linguistic corpus [18] from excerpts of several articles, short stories and scientific reports. Together, it mixed distinct topics and types of narrative, with a considering number and diversity of events and temporal expressions. The corpus, manually annotated, featured about 100 temporal expressions, 700 events and 700 relations, from almost 200 distinct sentences and 4500 words.

There were 3 different modules needing evaluation: event detection (already in-built in the system), event relation and event sorting module. Considering that, the evaluation system ran through the solution in three stages:

- Firstly, it evaluated all results as a whole;
- Secondly, it removed every incorrect or missing event and dependencies related with them, looking at remaining results, evaluating the relational and sorting modules as one;
- Lastly, it removed all incorrect and missing events' relations, removing results caused by those, and evaluated the remaining output.

It as wise to also consider making a strict and a relaxed evaluation. The strict evaluation evaluates the output without allowing any faults while the relaxed evaluation allow some representation and minor discrepancies while comparing it with the expected outcome, like wrong classification of type in events or relations.

In the evaluation, each dependency could be considered *correct*, *incorrect* or *missing*, if it was not found and should have been. Combining these criteria, the modules could be classified through a confusion matrix and in terms of *recall*, *precision* and *f-measure*.

We can see the results in the next 2 tables, displaying an strict evaluation 1 and a relaxed evaluation 2.

Modules	Precision (%)	Recall (%)	F-Measure (%)
Total	24,4	8,1	12,2
By task:			
- Event Detection	77,8	68,4	72,8
- Temporal Relation	42,9	31,6	36,4
- Event Relation	52,3	31,1	39,0
- Event Sorting	51,9	25,6	44,3

Table 1. Results from strict evaluation.

Modules	Precision (%)	Recall (%)	F-Measure (%)
Total	24,4	8,1	12,2
By task:			
- Event Detection	77,8	68,4	72,8
- Temporal Relation	42,9	31,6	36,4
- Event Relation	69,0	37,4	48,5
- Event Sorting	51,1	35,0	41,6

Table 2. Results from relaxed evaluation.

In terms of graphical representation, it was achieved a lot of small graphs expressing the low recall of the solution. It was necessary to add an unique identifier for each event and temporal expression in the graph to ensure that distinct events, but identified with the same graphic word, were represented as the same event.

Among graphs of average size obtained from the event sorting it was generated graphs with a good solution and readability. However, it was detected a problem while representing of events concurrent among themselves. When there are many events occurring *simultaneously*, it becomes difficult to understand at first glance that those events occur at the same time

7 Discussion

The module already has a considerable accurancy and comprehensiveness in matching and sorting events. It obtained a low accuracy and coverage, but it can be explain by some decisions made throughout this work. Being the complex project , the task of matching relationship of events were simplified by the use of conjunctions, prepositions and adverbs identified. Therefore, there was not detected the relations between events when the event was in a subordinate relative clause [1]. In other situations, two events are linked between each other by complex prepositional phrases, making mor difficult the correct detection of relations. It was also found more errors and missing relations in excerpts with

[1] In the case of relative clauses there is not an explicit connector, because the relative pronoun functions as a constituent of the subordinate clause.

an high amount of delimiters of speech (like the colon). These delimiters enable the segmentation of large or complex sentences, but hinder their analysis and make it difficult to extract relations by the same methods used in simple sentences. Additionally, there are still some complex temporal expressions that are not properly identified, preventing the extraction of their relations, like ther term *genesíacos during the days of 26 and 27*.

The ordering of events depended largely on the interpretation of the meaning of connectors and events. The low recall obtained was due mostly about the difficulty of this task. When the sorting task involved ordering of events obtained from nominal constructions which lack temporal and aspetual information, it wasn't possible to sort them by any criteria studied in this work. And, if the event was tying verbal nature and possessed such information, some mistakes would be made without further semantics analysis.

With a more use of temporal expression, like using dates (eg: *3 April*, *end of 2001*) and other temporal references (eg: *Christmas*, *on Wednesday morning*) can help improving the solution.

8 Conclusion

As studied, the problem of identification and ordering of events is quite complex and extensive to be fully solved in a single step, so we reduced the set of relations to be treated and simplified the problem of identifying events. One of those simplifications refers to the fact of considering relationship only within a single sentence and, unlike some systems analyzed, considering sorting only local relations instead of considering all simultaneously. It was also simplified the work by considering only the sorting relations *before* and *simultaneously*. Still, the solution is structured in order to enable future conversion logic based on intervals, in accordance of the systems reviewed during this work . It was also expanded the identification of non-standardized events and develop a solution based on the analysis of connectors, temporal adverbs and verbs associated with events. When evaluating the solution, as described, it was detected are some situations that require more effort and time to resolve since the module demonstrates problems in the sorting events when there is little information is available, such as the presence of unmarked verb constructions or lack of connectors, and more complex problems, such as using composite connectors or sentences fragmented by commas.

The use of statistics or a probabilistic method can assist in resolving these situations since this approach has already been successfully applied in previous systems, particularly on TRIPS and TRIOS (2010) [9], NCSU-INDI (2010) [17] and Marşic (2011) [3]. However, it will be needed a corpus of large dimensions to train and test.

In this solution we used a rule-based, similar to TERSEO (2005) [4], and Marşic (2011) approach which requires a very careful analysis of the language and, with more investment in it, it is possible to obtain better results. As a suggestion, one improvement to this solution would be to invest in the interpretation

of temporal expressions, determining the textual and enunciation moment, and in the analysis of delimiters of speech.

Another situation, which other systems took into account, such as Chambers et al. (2008) and Marşic (2011), was the achievement of the overall analysis of the relationships between events. Chambers et al. performed various approaches and reported an overall increase in accuracy and scope to address the problem both locally and globally.

An alternative to rule-based approach involves a strategy of machine learning as in Chambers et al.(2008) [16], or a hybrid solution. Both require the use of a training corpus and large to obtain good test results, but proved rewarding the effort expended in its construction. The Marşic's system is a good example of this hybrid approach, using a rule-based methods in conjunction with machine learning or results of existing statistics in an attempt to combine the best of both worlds approach.

Bibliography

[1] Mamede, N., Baptista, J., Diniz, C., Cabarrão, V.: STRING: An Hybrid Statistical and Rule-Based Natural Language Processing Chain for Portuguese in Propor 2012, Demo (2012)

[2] Casati, R., Varzi, A.: Events. In Zalta, E.N., ed.: The Stanford Encyclopedia of Philosophy. Spring 2010 edn. (2010)

[3] Marşic, G.: Temporal Processing of News: Annotation of Temporal Expressions, Verbal Events and Temporal Relations. PhD thesis, University of Wolverhampton, Wolverhampton, UK (2011)

[4] Saquete, E., Muñoz, R., Martínez-Barco, P.: TERSEO: Temporal Expression Resolution System Applied to Event Ordering. In Matoušek, V., Mautner, P., eds.: Text, Speech and Dialogue. Volume 2807 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2003) 220–228

[5] Fellbaum, C., ed.: WordNet: An Electronic Lexical Database (Language, Speech, and Communication). illustrated edition edn. The MIT Press, Cambridge, MA (1998)

[6] Miller, G.A.: WordNet: A Lexical Database for English. In: Communications of the ACM. Volume 38. (1995) 39–41

[7] Harabagiu, S.M., Miller, G.A., Moldovan, D.I.: Wordnet 2 - a morphologically and semantically enhanced resource. In: University of Maryland. (1999) 1–8

[8] Chklovski, T., Pantel, P.: VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In Lin, D., Wu, D., eds.: Proceedings of EMNLP 2004, Barcelona, Spain, Association for Computational Linguistics (2004) 33–40

[9] UzZaman, N., Allen, J.F.: TRIPS and TRIOS System for TempEval-2: Extracting Temporal In-formation from Text. (2010) 276–283

[10] Hagège, C., Tannier, X.: XRCE-T: XIP temporal module for TempEval campaign. (2007) 492–495

[11] Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: Robust Specification of Event and Temporal Expressions in Text. (2003)

[12] Saurí, R., Pustejovsky, J.: TimeML in a Nutshell. (2009)

[13] TimeML Working Group: Guidelines for Temporal Expression Annotation for English for TempEval 2010. (2009)

[14] Saurí, R., Goldberg, L., Verhagen, M., Pustejovsky, J.: Annotating Events in English TimeML Annotation Guidelines. (2009)

[15] Allen, J.F., Ferguson, G.: Actions and Events in Interval Temporal Logic. (1994)

[16] Chambers, N., Jurafsky, D.: Jointly Combining Implicit Constraints Improves Temporal Ordering. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, Hawaii, Association for Computational Linguistics (2008) 698–706

[17] Ha, E.Y., Baikadi, A., Licata, C., Lester, J.C.: NCSU: Modeling Temporal Relations with Markov Logic and Lexical Ontology. (2010) 341–344
[18] Garside, R., Leech, G.N., McEnery, T., Others: Corpus annotation: linguistic information from computer text corpora. Longman (1997)