Event Ordering through Temporal Expression Resolution*

E. Saquete, R. Muñoz and **P. Martínez-Barco**
Grupo de investigación del Procesamiento del Lenguaje y Sistemas de Información.
Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante.
Campus de San Vicente
San Vicente del Raspeig 03690, Spain
{stela,rafael,patricio}@dlsi.ua.es

Abstract

In this paper a multilingual method for event ordering based on temporal expression resolution is presented. This method has been implemented through the TERSEO system which consists of three main units: temporal expression recognizing, resolution of the coreference introduced by these expressions, and event ordering. By means of this system, chronological information related to events can be extracted from documental databases. This information is automatically added to the documental database in order to allow its use by question answering systems in those cases referring to temporality. The system has been evaluated obtaining results of 91 % precision and 71 % recall. For this, a blind evaluation process has been developed guaranteing a reliable annotation process that was measured through the *kappa* factor.

1 Introduction

Nowadays, the information society needs a set of tools for the more and more increased amount of digital information stored in Internet. Documental database applications help us to manage this information. However, documental database building requires the application of automatic processes in order to extract relevant information from texts.

One of these automatic processes is the event ordering by means of the date and time. Usually, a user needs to obtain all the information related to a specific event. To do this, he must know the relationships between other events, and their chronological information. The automatic identification of temporal expressions associated to events, and further treatments of them, allows to build their chronographic diagram. Temporal expressions treatment is based on establishing relationships between concrete dates or time expressions (25th December 2002) and relative dates or time expressions (the day before). After that, the application of event-ordering techniques allows to obtain the desired event ordering.

This paper has been structured in the following way: first of all, section 2 shows a short introduction of the main contributions of previous work. Then, section 3 presents a deep study extracted from a training corpus allowing to establish a taxonomy of temporal expressions. In section 4, the general architecture and description of the system used to identify and process temporal expressions is done. Moreover, the scores obtained by the system when it was evaluated against a test corpus are presented. In section 5, the application of this system together with an event ordering method in documental databases is introduced and some conclusions are shown.

2 Previous work

At the moment there are different kind of systems that try to annotate and resolve temporal expressions in different types of corpus:

- **Based on knowledge**. These systems have a previous knowledge base that contains the rules used to solve the temporal expressions.
- **Based on Machine Learning**. In this kind of systems, a supervised-annotated corpus is needed to automatically generate the system rules that can have a percentage of appearance of these rules in corpus. On these rules the system is based.

Within the ones based on knowledge there are works like Wiebe et al.(Wiebe *et al.* 98), that use a set of defined rules. For the current temporal expression to be resolved, each rule is applied. For the anaphoric rules, the antecedent considered is the most recent one satisfying the constraints. All consistent maximal mergings of the results are formed, and the one with the highest score is the chosen interpretation. However, the corpora used in this system are scheduling dialogs, in which temporal expressions are limited.

* This paper has been supported by the Spanish government, projects FIT-150500-2002-244 and FIT-150500-2002-416.

Filatova and Hovy (Filatova & Hovy 01) is another system based on knowledge that describes a method for breaking news stories into their constituent events and assigning time-stamps to them. Schilder and Habel (Schilder & C.Habel 01) system is knowledge based as well, however, it is only resolving expressions that refer to the article date and not the ones that refer to a previous date of the text. By contrast, some of the most important systems based on Machine Learning are, for instance, Wilson et al.(Wilson *et al.* 01), Katz and Arosio (Katz & F.Arosio 01), Setzer and Gaizauskas (Setzer & R.Gaizauskas 02). This last one focuses on annotating Event-Event Temporal Relations in text, using a time-event graph which is more complete but costly and error-prone either.

Although with the systems based on Machine Learning high precision results are obtained within the domain in which they have been trained, these results are lower when this kind of systems are applied to other domains. Besides, it needs large annotated corpus. On the other hand, systems based on knowledge have a greater flexibility to be applied to any domain. In spite of the existence of multiple methods of TEs resolution, considering that this task depends on the language, actually there are not complete systems that resolve this task in Spanish. For this reason, firstly, a study of the different types of TEs in Spanish has been made, as well as the study of the possible resolution strategies. This study is presented in the following section.

3 Taxonomy of temporal expressions

One of the main tasks trying to recognize and resolve temporal expressions is to classify them, because the way of solving it depends on the type of the expression. In this paper, two proposals of classification of temporal expressions are shown as well as an interrelation between them using some examples. The first classification is based on the kind of reference. This classification is focused on recognizing the kind of expression when this enters the system and needs to be resolved.On the other hand, another type of classification is presented as well. This one is focused on the kind of output returned by the system for that type of expression.

3.1 Classification of the expression based on the kind of reference

- **Explicit Temporal Expressions**.
  - *Complete Dates with or without time expressions*: "11/01/2002" (01/ 11/2002), "el 4 de enero de 2002" (January 4th,2002),...
  - *Dates of Events:*
    - Noun Phrase with explicit date: "el curso 2002-2003" (2002-2003 course). In this expression, "course" denotes an event
    - Noun Phrase with a well-known date: "Navidad" (Christmas),...
- **Implicit Temporal Expressions**.
  - Exp. that refer to the *Document date*:
    - Adverbs or adverbials phrases: "ayer" (yesterday),...
    - Noun phrases:"el próximo mes" (the next month),...
    - Prepositional phrases: "en el mes pasado" (in the last month),...
  - Exp. that refers to *another date*:
    - Adverbial Phrases:
      - Simple Expressions: "durante el curso" (during the course),...
      - Complex Expressions: "anteayer" (the day before yesterday), "pasado mañana" (the day after tomorrow),...For example, in this last expression it is necessary to resolve "tomorrow" first and then resolve "after" applied to the date obtained previously.
    - Noun Phrases
      - Simple Expressions: "un mes después" (a month later), "un día antes" (a day before),...All these expressions are referring to a date named before in the text.
      - Complex Expressions:"después de la próxima Navidad" (after next Christmas),...For example, with the expression "after next Christmas" it is necessary to resolve the TE "next Christmas" and then apply the changes that the word "after" makes on the date obtained.
    - Prepositional Phrases

· Simple Expressions : “desde Navidad” (from Christmas),...
· Complex Expressions: “desde la anterior Navidad” (since last Christmas),...

3.2 Classification by the representation of the temporary value of the expression

- **Concrete**. All those that give back a concrete day or/and time with format: dd/mm/yyyy (hh:mm:ss), for example: “yesterday”.

- **Period**. All those expressions that give back to a time interval or range of dates: [dd/mm/yyyy-dd/mm/yyyy], for example: “during the five following days”.

- **Fuzzy**. It gives back an approximate time interval because it exactly does not know the concrete date to which the expression talks about. They can be of two types:

  – *Fuzzy concrete.* If the given back result is an interval but the expression is referring to a concrete day within that interval, and we do not know which is exactly. For that reason we must give back the approach of the interval, for example: “un día de la semana pasada” (a day of the last week),...
  – *Fuzzy period.* If the expression talks about an interval contained within the given back interval, for instance:“hace unos días” (some days before), “durante semanas” (during weeks),...

In addition, we can relate both types of classifications with different expressions, for example:

- **“ayer”** (yesterday). Expression of implicit date, that makes reference to the date of the document,adverbial and concrete.

- **“el mes pasado”** (the last month).Expression of implicit date, that makes reference to the date of the document, that is noun phrase and that gives back a period.

- **“antes del próximo curso”** (before the next course) .It would be an anaphoric expression of implicit date that makes reference to an event that in addition would be of the fuzzy type.

In the following section we will see how the system is able to recognize and solve great part of these temporal expressions.

4 Description of TERSEO system

TERSEO (Temporal Expression Resolution System applied to Event Ordering). In Figure 1 the graphic representation of the system proposed for the recognition of TEs and for the resolution of its references is shown, according to the temporal model proposed. The texts are tagged with lexical and morphological information and this information is the input to the temporal parser. This temporal parser is implemented using an ascending technique (chart parser) and it is based on a temporal grammar. Once the parser recognizes the TEs in the text, these are introduced into the resolution unit, which will update the value of the reference according to the date it is referring and generate the XML tags for each expression. Finally, these tags are the input of a event ordering unit that gives back the ordered text. We can find explicit and implicit TEs. The grammar in Tables 1 and 2 is used by the parser to discriminate between them.

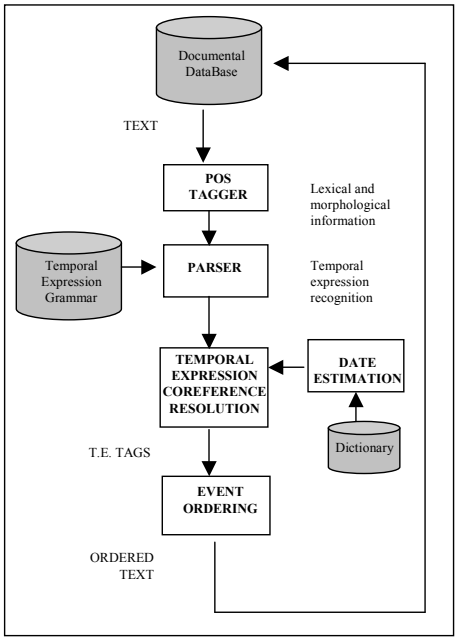

Figure 1: Graphic representation of TERSEO

4.1 Temporal expression parser module

The parser uses a grammar based on two different kinds of rules. On one hand there are rules for the date and time recognition (Explicit dates 12/06/1975) and on the other hand there are rules for the temporal reference recognition (implicit dates TEs that need the location of another complete TE to be understood "two days before"). The grammar proposed recognize the large amount of date and time formats. In Table 1 a sample of some of the rules that have been defined in our system for the date recognition can be observed[1].

fecha→ dd+'/'+mm+'/'+(yy)yy	(12/06/1975)
	(06/12/1975)
fecha→ dd+'de'+mm+'de'+(yy)yy	
	(12 de junio de 1975)
	(12th of June of 1975)
fecha→ ('El')+diasemana+dd+'de'+mes+'de'+(yy)yy	
	(El domingo 12 de junio de 1975)
	(Sunday, 12th of June of 1975)
hora→ hh+':'+mm+(':'+ss)	*(time)*

Table 1: Sample of rules for Explicit Dates Recognition

There are two types of temporal references that should be treated: the time adverbs (i.e. yesterday, tomorrow) and the nominal phrases that are referring to temporal relationships (i.e. the day after, the day before). In Table 2 we show some of the rules used for the detection of every kind of reference.

4.2 Anaphoric relation resolution module based on a temporal model

For the anaphoric relation resolution we use an inference engine that interprets every reference named before. In some cases the references are estimated using the newspaper's date (FechaP). Others refer to a date named before in the text that is being analyzed (FechaA). For these cases, a temporal model that allows to know over what date the dictionary operations are going to be done, is defined. This model is based on the two rules below and it is only applicable to these dates that are not FechaP, since for FechaP there is nothing to resolve:

1. By default, the newspaper's date is used as a base referent (TE) if it exits, if not, the system date is used.

2. In case of finding a non-anaphoric TE, it is stored as FechaA. This value is updated every time that a non-anaphoric TE appears in the text.

In Table 3 some of the entries of the dictionary used in the inference engine are shown.What is not a trivial task is to solve fuzzy TEs. Fuzzy TEs are not associated to a concrete date or time expression, however we can estimate a period of time in which the fuzzy TE could have occurred. For example, "a day of the last week" has not a concrete day, but we could estimate this day occurs between 23 December 2002 and 29 December 2002 if the document date is 31 December 2002.

4.3 Tagging of temporal expressions

Several proposals for the annotation of TEs have been arisen in the last few years Wilson et al.(Wilson *et al.* 01), Katz and Arosio (Katz & F.Arosio 01) since some research institutions have started to work on different aspects of temporal information. However there is not a consensus on what and how temporal information should be identified in text. In this section, a set of XML tags is defined in order to standardize the different kinds of TEs. Our proposal is driven by a dual motivation: a) to reflect the results of our temporal-reference resolution system and b) to standardize the date-time format in Internet texts. These tags show the following structure:

- For explicit dates:

```
<DATE_TIME ID="value" TYPE="value"
VALDATE1="value" VALTIME1="value"
VALDATE2="value" VALTIME2="value"
VALORDER="value"> expresión </DATE_TIME>
```

- For implicit dates:

```
<DATE_TIME_REF ID="value" TYPE="value"
VALDATE1="value" VALTIME1="value"
VALDATE2="value" VALTIME2="value"
VALORDER="value"> expresión </DATE_TIME_REF>
```

DATE_TIME is the name of the tag for explicit TEs and DATE_TIME _REF is the name of the tag for implicit TEs. Every expression has an numeric ID to identify it and VALDATE# and VALTIME# store the range of dates and times

[1]The nomenclature in which the rules are abovementioned is the one used in the data dictionaries of the information systems. Words between brackets are optional, words between square brackets, [ ] mean that only one of them appear in the rule and words between keys, { } will be repeated one or more times in the rule.

Implicit dates referring to Document Date Concrete	`referencia→ 'ayer'` *(yesterday)* `referencia→ 'mañana'` *(tomorrow)* `referencia→ 'anteayer'` *(the day before yesterdary)* `referencia→ 'el próximo día'` *(the next day)*
Implicit Dates Previous Date Period	`referencia→ 'un mes después'` *(a month later)* `referencia→ num+'años después'` *(num years later)*
Imp. Dates Prev.Date Concrete	`referencia→ 'un día antes'` *(a day before)*
Implicit Dates Previous Date Fuzzy	`referencia→ 'días después'` *(some days later)* `referencia→ 'días antes'` *(some days before)*

Table 2: Sample of rules for Implicit Dates recognition

REFERENCE	DICCIONARY ENTRY
'ayer' *(yesterday)*	`Day(FechaP)-1/Month(FechaP)/Year(FechaP)`
'mañana' (tomorrow)	`Day(FechaP)+1/Month(FechaP)/Year(FechaP)`
'anteayer' *(the day before yesterday)*	`Day(FechaP)-2/Month(FechaP)/Year(FechaP)`
'el próximo día' *(the next day)*	`Day(FechaP)+1/Month(FechaP)/Year(FechaP)`
'un mes después' *(a month later)*	`[DayI/Month(FechaA)+1/Year(FechaA)-- DayF/Month(FechaA)+1/Year(FechaA)]`
num+'años después' *(num years later)*	`[01/01/Year(FechaA)+num -- 31/12/Year(FechaA)+num]`
'un día antes' *(a day before)*	`Day(FechaA)-1/Month(FechaA)/Year(FechaA)`
'días después' *(some days later)*	`>>>>FechaA`
'días antes' *(some days before)*	`<<<<FechaA`

Table 3: Sample of some of the entries of the dictionary

obtained from the inference engine, where VALDATE2 and VALTIME2 is only used to establish ranges. Also, VALTIME1 could be omitted if only a date is specified. VALDATE2, VALTIME1 and VALTIME2 are optional args. VALORDER is the attribute where the event ordering unit will put the value, at first there is no value for this attribute. After that, a structured document is obtained. The use of XML allows us to take advantage of the XML schema in which the tag language is defined. This schema lets an application know if the XML file is valid and well-formed. A parser of our XML needs to be defined to make the information useful.

The unit estimating the dates will accede to the right entry in the inference engine in each case and it will apply the function specified obtaining a date in the format dd/mm/yyyy or a range of dates. So, at that point the anaphoric relation will have been resolved.

4.4 Event ordering module

Event ordering in natural language written texts is not a trivial task. Firstly, a process to identify events must be done. Then, the relationship between the event and the date when the event occurs must be identified. Finally, the ordering of events must be determined according to their estimated dates. As our method is not an event identification method but an event ordering, we have trivialized the task of identifying events and their relation to a TE. In this system, an event is considered as the sentence that includes some kind of TE. So, in our method, we have reduced this task to the identification of TEs (*concrete*, *periodo* or *fuzzy*, according to the above-mentioned classification). After that, the process to temporal expression resolution must be performed in order to a) solve implicit TEs ; and b) add XML date-time tags for explicit and implicit TEs.

Once XML tags are included in the text, the event ordering module runs over the text building a matrix with the complete information from the XML tags. This matrix includes the columns ID, VALDATE1, VALTIME1, VALDATE2, VALTIME2, and VALORDER (empty at the moment). Once the matrix has been built the module sorts it in ascendent order according to the following rules:

1. TE1 is previous to TE2, if the range of VALDATE1, VALTIME1, VALDATE2, VAL-

TIME2 associated to TE1 is prior to and not overlapping the range associated to TE2.

2. TE1 is concurrent to TE2, if the range of VALDATE1, VALTIME1, VALDATE2, VALTIME2 associated to TE1 overlaps the range associated to TE2.

After that, the system will assign a sequential order number to every TE in the matrix, having the same order number for concurrent TEs. Then, the document is re-tagged including this order number in the VALORDER attribute for every XML tag.

Next, a example is shown. Newspaper's date: 30/12/2002

```
[El miércoles 13 de noviembre de 2002]
<DATE_TIME VALDATE1=‘‘13/11/2002’’ VALORDER =
‘‘2’’>, el barco Prestige lanza un SOS a 28
millas de Finisterre debido a una brecha. El
barco había sido parcheado [unos meses antes]
<DATE _TIME_REF VALDATE1 = ‘‘<<<<13/11/2002’’
VALORDER = ‘‘1’’> en un astillero chino. Al
conocerse la noticia, la Agencia Marítima y
de Guardacostas del Reino Unido ofrecen su
ayuda al estado español para controlar la
contamina- ción, pero ésta no será aceptada
hasta [el 22 de noviembre] <DATE_TIME VALDATE1
= ‘‘22/11/2002’’ VALORDER = ‘‘5’’>. [El 15 de
noviembre]<DATE_TIME VALDATE1 = ‘‘15/11/2002’’
VALORDER=‘‘3’’> comienzan a acercarse manchas
y [el 16 de noviembre] <DATE_TIME VALDATE1
= ‘‘16/11/2002’’ VALORDER = ‘‘4’’> 190
kilometros de costa se ven anegados por la
marea negra. [Seis días después] <DATE_TIME_REF
VALDATE1 = ‘‘22/11/2002’’ VALORDER = ‘‘5’’>,
Manuel Fraga visita por primera vez la zona
afectada. [El 1 de diciembre] <DATE_TIME
VALDATE1 = ‘‘01/12/2002’’ VALORDER=‘‘7’’>, el
basticafo francés Nautilus llega a la costa
gallega. [Previamente] <DATE_TIME_REF VALDATE1
=‘‘<<<<01/12/2002’’ VALORDER=‘‘6’’>, en el
Prestige ya se habían producido nuevas grietas.
```

(*[Wednesday 13 of November of 2002], the Prestige boat sends a SOS to 28 miles of Finisterre due to a breach. The boat had been patched [months before] in a Chinese shipyard. When knowing itself the news, the Marine Agency and Coastguard vessel of the United Kingdom offers their aid to the Spanish state to control the contamination, but this one will not be accepted [until the 22 of November]. [The 15 of November] spots begin to approach and the [16 of November] 190 kilometres of coast are flooded by the black tide. [Six days later], Manuel Fraga visits the affected zone for the first time. [The 1 of December], the French bathyscaphe Nautilus arrives at the Galician coast. [Previously], in the Prestige already new cracks had taken place.*)

5 System evaluation

In order to make the evaluation of this system, a manual annotation of texts has been made previously by two annotators with the purpose of comparing it with the automatic annotation that produces the system. For that reason, it is necessary to assure that the manual information is trustworthy and it does not alter the results of the experiment. Carletta (Carletta et al. 97) explains that to assure a good annotation is necessary to make a series of direct measurements that are: stability, reproducibility and precision, but in addition to these measurements the reliability must measure the amount of noise in the information. The authors argue that due to the amount of agreement by chance that one can expected depends on the number of relative frequencies of the categories under test, the reliability for the classifications of categories would have to be measured using the factor *kappa* defined in Siegel and Castellan (Siegel & Castellan 88). *Kappa (k)* measures the affinity in agreement between a set of annotators when they make categories judgments.

In our case, there is only one class of objects and there are three objects within this class: objects that are referring to the date of the article, objects which are referring to the previous date and objects that are referring to another date different from the previous ones.

After carrying out the calculation, a value k=0.953 has been obtained. According to the work of Carletta (Carletta et al. 97), a measurement of k like 0,68 $< k <$ 0,8 means that the conclusions are favorable, and if $k >$ 0,8 means that exists total reliability between the results of both annotators. Since our value of k is greater than 0,8, it is guaranteed that a total reliability in the conducted annotation exists and therefore, the results of obtained precision and recall are guaranteed.

An evaluation of the module of resolution of TEs has been made. Two corpus formed by newspaper articles in Spanish has been used. The first set has been used for the training and it consists of 50 articles. Thus, after making the opportune adjustments to the system, the optimal results of precision and recall were obtained that are in Table 4.

	TRAINING	**TEST**
N*o* Art.	50	50
Real Ref	238	199
Treated Ref.	201	156
Successes	170	138
Precision	84.58%	88.46%
Recall	71.43%	69.35%

Table 4: Evaluation of the system

Although the obtained results are highly successful, we have detected some fails that have been deeply analyzed. As it can be observed by the results, our system could be improved in some aspects. Below, a study of the problems detected and their possible improvements is shown:

In the newspaper's articles, sometimes there are expressions like *"el sábado hubo cinco accidentes"* (Saturday there were five accidents). To resolve this kind of references we should need context information of the sentence where the reference is. That information could be the time of the sentence's verb. If the verb is a past verb, it indicates that it is necessary to solve a reference like *"el sábado pasado"* (last Saturday), whereas if it is a future verb it refers to *"el sábado próximo"* (the next Saturday). Because of our system does not use semantic or context information we assume this kind of reference is referring to the last day, not the next, because the news usually tells us facts occurred previously.

At the moment our system solves the anaphoric expressions that make reference to the date of the article or to another explicit date named previously in the text, but is possible that some expressions could make reference to dates which have been obtained solving another previous reference.These cases are not being controlled by our system, and the system is returning a bad solution. For example: *"there was an accident two days ago. Three days later..."*. In this case, the second expression *"Three days later"* makes reference to the date previously calculated by the expression *"two days ago"* and not to a explicit date named before.

6 Application and conclusions

Users of Documental Database sometimes need to know all the facts related to an event. For this reason, it is useful to have tools in order to extract all the information about the event for establishing a chronology of events. The treatment of temporal expressions with the ordering method makes up the main modules of our tools. Moreover, the chronological tool is made up of a event recognition module in order to extract the event related to the news.

Following, a short description for the full system is shown focusing on extracting the events related to news from the documental database:

1. First of all, the title of the news is linked to the date of the documents.
2. Secondly, the system removes all sentences without temporal expressions. We only consider as event the sentence in which a TE appears.
3. After that, the module for treatment of temporal expressions is applied according to the previous explanation.
4. Finally, once the coreference module is applied, the ordering module is ran tagging the order of the events in the text.

This system can help Question Answering systems to provide the information to users when asking for questions like *What happened the day before?* or *What happened the last day?* (Pustejovsky 02).

Nowadays, the application of machine learning techniques (i.e. genetic algorithms) to the automatic extraction of temporal rules for new languages is being performed. Also, the adaptation and comparison with the standard TIME-ML (Pustejovsky 03) arisen from recent specific workshops is being studied.

References

(ACL 01) ACL, editor. *Proceedings of the 2001 ACL-EACL, Workshop on Temporal and Spatial Information Processing*, Toulouse,France, 2001.

(Carletta et al. 97) Carletta et al. The Reliability of a Dialogue Structure Coding Scheme. *Computational Linguistics*, 23(1):13–32, 1997.

(Filatova & Hovy 01) E. Filatova and E. Hovy. Assigning Time-Stamps to Event-Clauses. In ACL (ACL 01), pages 88–95.

(Katz & F.Arosio 01) G. Katz and F.Arosio. The Annotation of Temporal Information In Natural Language Sentences. In ACL (ACL 01), pages 104–111.

(Pustejovsky 02) J. Pustejovsky. Time and event recognition for question answering systems. www.cs.brandeis.edu/ jamesp/arda/time/terqas, 2002.

(Pustejovsky 03) J. Pustejovsky. Timeml markup language for temporal and event expressions. www.cs.brandeis.edu/ jamesp/arda/time, 2003.

(Schilder & C.Habel 01) F. Schilder and C.Habel. From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages. In ACL (ACL 01), pages 65–72.

(Setzer & R.Gaizauskas 02) A. Setzer and R.Gaizauskas. On the Importance of Annotating Event-Event Temporal Relations in Text. In LREC, editor, *Proceedings of the LREC Workshop on Temporal Annotation Standards, 2002*, pages 52–60, Las Palmas de Gran Canaria,Spain, 2002.

(Siegel & Castellan 88) S. Siegel and J. Castellan. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, 2nd edition, 1988.

(Wiebe *et al.* 98) J.M. Wiebe, T.P. O'Hara, T. Ohrstrom-Sandgren, and K.J. McKeever. An Empirical Approach to Temporal Reference Resolution. *Journal of Artificial Intelligence Research*, 9:247–293, 1998.

(Wilson *et al.* 01) G. Wilson, I. Mani, B. Sundheim, and L. Ferro. A Multilingual Approach To Annotating And Extracting Temporal Information. In ACL (ACL 01), pages 81–87.