Undefined 0 (2019) 1–0
IOS Press

Annotador: a Temporal Tagger for Spanish

Abstract. Temporal information is crucial in knowledge extraction. Being able to locate events in a timeline is necessary to understand the narrative behind every text. To this aim, several temporal taggers have been proposed in literature – nevertheless, not all languages received the same attention. Most taggers work only for English texts, and not many have been developed for other languages. Also the scarcity of annotated corpora in other languages notably hinders the task. In this paper we present a new rule-based tagger called *Annotador* (*Añotador* in Spanish) able to process texts both in Spanish and English. Furthermore, a new corpus with more than 300 short texts containing common temporal expressions, called the HourGlass corpus, has been built in order to test it and to facilitate the development of new resources and tools. Professionals from different domains intervened in the gathering of the text, making it heterogeneous and easy to use thanks to the tags added to each entry. Finally, we analyzed main challenges in the time expression extraction task.

Keywords: Time Expression, Temporal Tagger, Spanish language, NLP

1. Introduction

A *temporal tagger* is a system that extracts temporal expressions from texts and recognizes their meaning. *Time expressions* (also known as *temporal expressions* or *TEs*) are "constructions referring to points or intervals on the timeline" [1], and in general they can be understood as *anything that answers the questions 'when' or 'how long' but does not involve an event* (e.g. "2 May 2019" or "one hour"). Temporal taggers must first identify the time expressions (*identification*), and then resolve (*normalization*) their meaning, obtaining a fixed date from expressions such as "tomorrow". Table 1 shows some examples of normalization.

Table 1
Examples of normalization for several expressions using as reference date (*anchor date*) December 20, 2019 (Friday).

#	Spanish Expression	English	Normalized Value
1	mañana	tomorrow	2019-12-21
2	el mes que viene	the next month	2020-01
3	el pasado lunes	last Monday	2019-12-16

This paper presents *Annotador* (from the Spanish *Añotador*[1]), a temporal tagger for Spanish and English –although we will focus in Spanish in the scope of this paper. Annotador was conceived for the automatic creation of timelines from legal documents, but it is of general purpose[2]. Annotador is part of the suite of services offered by the H2020 **ANONYMIZED FOR REVIEWERS** project for both English and Spanish texts.

Temporal tagging is a well recognized NLP task, and as such, still imperfect –different challenges periodically used appear in events (e.g. SemEval), where state-of-the-art temporal taggers, such as HeidelTime [2] and SUTime [3], show their performance. There are also specifications for representing temporal expressions, such as the TIMEX3 tags, which are part of the markup language TimeML [4].

However, Annotador has been deemed necessary due to the low coverage shown by temporal taggers of Spanish texts –which are anyway scarce. In order to evaluate Annotador, one existing corpus is used, but as it will be reviewed in Section 4, existing corpora fall short of rich features and availability; therefore a new corpus is presented.

The contributions of this paper are the following: (i) a temporal tagger for Spanish texts that outperforms existing libraries and (ii) a new test bed called HourGlass, a collection of temporal expressions where both synthetic texts and contributions by volunteers foreign to temporal tagging were gathered, analyzed and annotated.

The remainder of this paper is structured as follows: Section 2 describes previous work on temporal information extraction and annotated corpora for Spanish. Section 3 presents Annotador, our new temporal tagger

[1] It is a pun: "Año" means "Year" and "Anotador" is the person or tool that annotates something.

[2] Texts in the legal domain sometimes present a narrative character, and narrative threads can be found in police reports, cross examinations, consumer complaints, charges of indictment or legal appellate judgments; but the same narrative character appears in other documents such as clinical stories or newspaper articles.

for Spanish. Section 4 introduces the HourGlass corpus, explaining how it was gathered, categorized and annotated. Section 5 details the evaluation of our temporal tagger with regard to other state-of-the-art tools over documents from (1) the SemEval 2010 challenge TempEval 2[3] and (2) from our HourGlass corpus. Section 6 analyzes existing challenges in time expression extraction and some particularities in the Spanish language. Finally, Section 7 presents our conclusions and discusses future work.

2. Related work

2.1. Temporal Taggers

Despite many temporal taggers can be found in literature, most of them are not maintained or no longer available. Even among the operative ones, to the best of the authors' knowledge, only three of them target Spanish texts. In this section, we will briefly introduce the different approaches to temporal tagging and the most widely used temporal taggers, paying special attention to the ones able to process Spanish texts.

Temporal taggers can be rule-based, use machine learning techniques or combine both approaches. Rule-based temporal taggers are the most common and tend to offer the best results, but they are also less flexible. Lack of versatility, errors in the rules or unexpected interactions among them can provoke errors in the temporal annotation. Additionally, they are also more difficult to adapt to a new domain and language. Among rule-based taggers we find HeidelTime [5] and SUTime [3], both capable of processing Spanish texts to some extent, and also English only temporal taggers such as SynTime [6], a tool that defines different types of tokens (*time tokens, modifiers* and *numerals*) and identifies time expressions using rules based on these tokens (although it does not normalize them).

On the opposite, machine-learning-based temporal taggers are much more resilient to different ways to express time expressions, but demand big and variate corpora in order to train a powerful model. We can find among these ClearTK-TimeML [7], a tagger that identifies time expressions and classifies them in types (but does not normalize them) using external machine learning tools. Finally, among hybrid approaches (using both rules and machine learning techniques) we can find USFD2 [8], UWTime [9] (a context-dependent semantic parser that uses a Combinatory Categorial Grammar parser for identification and normalization of time expressions), TARSQI [10] (a system that incorporates GUTime [11], which uses C4.5 algorithm to automatically discover new rules from a set of manually created ones), CAEVO [12] (a sieve-based architecture that uses different classifiers in a pipeline, some of them rule-based and some of them using machine learning) and TIPSem [13], able to process Spanish.

Among the available taggers capable to process Spanish documents we find HeidelTime [5], a domain-sensitive rule-based temporal tagger available for more than 200 languages; although, only 13 of them are based on manually created resources, the other ones were automatically generated. From the hybrid approach, TIPSem [13] uses Semantic Role Labeling and Conditional Random Field models with semantic, morphological and synthactic consideration features. Finally, SUTime [3] is a temporal tagger built on TokenRegex [14] included in the NER annotator of the Stanford CoreNLP framework [15]. While both TIPSem and HeidelTime have specific resources to process English and Spanish, SUTime on the other hand was mainly built for English, although it also includes Spanish rules for time expression extraction and normalization.

2.2. Corpora

Regarding corpora, different datasets have been released in challenges, such as previously mentioned TempEval, or proposed in literature. A more thorough exploration of these corpora reveals that not only the ISO standard TimeML [4] is used to annotate the expressions, but also other formats, such as TIDES TIMEX2 [16], or simply variations of TimeML, such as the medical extension done for the THYME project [17].

If we analyze available corpora[4], we find that some domains and types of text received more attention than others. Most corpora are built from news (e.g., the Timebank corpus [19], the TempEval challenges datasets and the MEANTIME corpus [20]). Historical texts and medical texts, like the Wikiwars corpus [21] and the THYME corpus [17] respectively, have

[3] `http://semeval2.fbk.eu/semeval2.php?location=tasks&taskid=5`

[4] In this section we restricted to corpora annotated with time expressions. For more information on corpora annotated with events, a recent state of the art was done by Sprugnoli and Tonelli [18].

also been annotated in the past. Regarding language registers, we can find corpora with scientific abstracts [5], tweets [22] and colloquial texts [5]. Nevertheless, all the previously mentioned corpora (except of the MEANTIME and TempEval datasets, that are multilingual) are exclusively composed of English texts.

Spanish corpora are scarce, and to the best of the authors' knowledge, there are only three datasets available for this language with TIMEX3 tags. The Spanish TimeBank corpus[5] (with news and fiction texts), the ModeS TimeBank 1.0[6] (texts from the 17th and 18th centuries) and the MEANTIME corpus (news). There were also Spanish challenges in TempEval 2 and TempEval 3 competitions, but they are built on texts from a task-adapted fragment of TimeBank; additionally, the latter's test dataset is not available online anymore. The Spanish available corpora is therefore scarce and not heterogeneous, notably hindering the temporal tagging task in this language.

3. Annotador Tagger

Annotador is a rule-based system that operates over a Stanford CoreNLP pipeline[7]. This pipeline includes a tokenizer, a sentence splitter, a lemmatizer, a POS tagger, a Named Entity Recognition tagger and the TokensRegex [23], a framework for defining cascaded patterns over token sequences where we input our customized rules for time expression recognition. While for the English version of our tool we use the default models that CoreNLP offers, for Spanish we substituted the default lemmatizer and the POS tagger by the IXAPipes models[8] trained with the Perceptron on the Ancora 2.0 corpus [24]. Our rules are therefore applied on the output of the previous annotators at the last stage of the CoreNLP pipeline. Then, our normalization algorithm decides the value of each of the expressions detected by the rules and outputs them in the desired format (TIMEX3 or JSON).

Figure 1 depicts the pipeline of Annotador. Annotador requires an input text and optionally an *anchor date*. The *anchor date* is the date with regard the normalization will be done. This is, if our anchor date is "2019-05-20" and we find the expressions "the month of March" or "last December", the normalization will be "2019-03" and "2018-12", respectively (see Table 1 for more examples).

In Section 3.1 we present the rules we developed for TE identification; in Section 3.2 we introduce our normalization algorithm.

3.1. Rules

Annotador relies on a set of more than 200 iterative rules. These rules are token-based; this is, they take into account tokens instead of strings –this allows to consider information such as POS tagging or lemmatization. These rules are applied via the Stanford CoreNLP TokensRegex, where we find different types of rules. In our system we use namely the following:

- Tokens rules: they work on the token level and are applied at different stages, relying on information tagged by previous annotators in the pipeline (such as lemmas or POS) or previous rules (e.g., for the expression "two days", the rules would first of all annotate that "two" is a number and that "days" is a type of temporal *granularity*[9], and in a subsequent stage it would be able to detect all the temporal expressions compound by the sequence "number + some granularity").
- Composite rules: differently than tokens rules, that are applied just once each, these rules work iteratively on tokens rules and on previous composite rules until there are no more matches.

These rules may produce several different actions, namely annotations (this is, internal tags that will be used by subsequent rules) and results (the final expressions with a specific set of values that will be returned to the normalization algorithm for further normalization). In our case, the values we return are (1) the type of expression (DATE, TIME, SET or DURATION[10]), (2) its normalized *value* (that might require further normalization or not), (3) the *freq* (in case it is a SET, oth-

[5] `https://catalog.ldc.upenn.edu/LDC2012T12`

[6] `https://catalog.ldc.upenn.edu/LDC2012T01`

[7] `https://stanfordnlp.github.io/CoreNLP/pipelines.html`

[8] To inject IXApipes into the Stanford CoreNLP pipeline, we adapted part of the code in `https://github.com/dhfbk/spanish`

[9] We call *granularity* to expressions such as "day", "month" or "century", that denote a specific way to measure periods of time.

[10] These are the types envisaged by the TimeML standard. DATE refers to calendar expressions such as "October", "December, 4 2019" or "the first quarter of the year". TIME covers clock expressions such as "one o'clock" or "tomorrow at 11pm". SETs are expressions that repeat over time, such as "monthly" or "two days a week". Finally, DURATION is the type denoting periods of time, such as "one week and a half" or "two days and three hours".

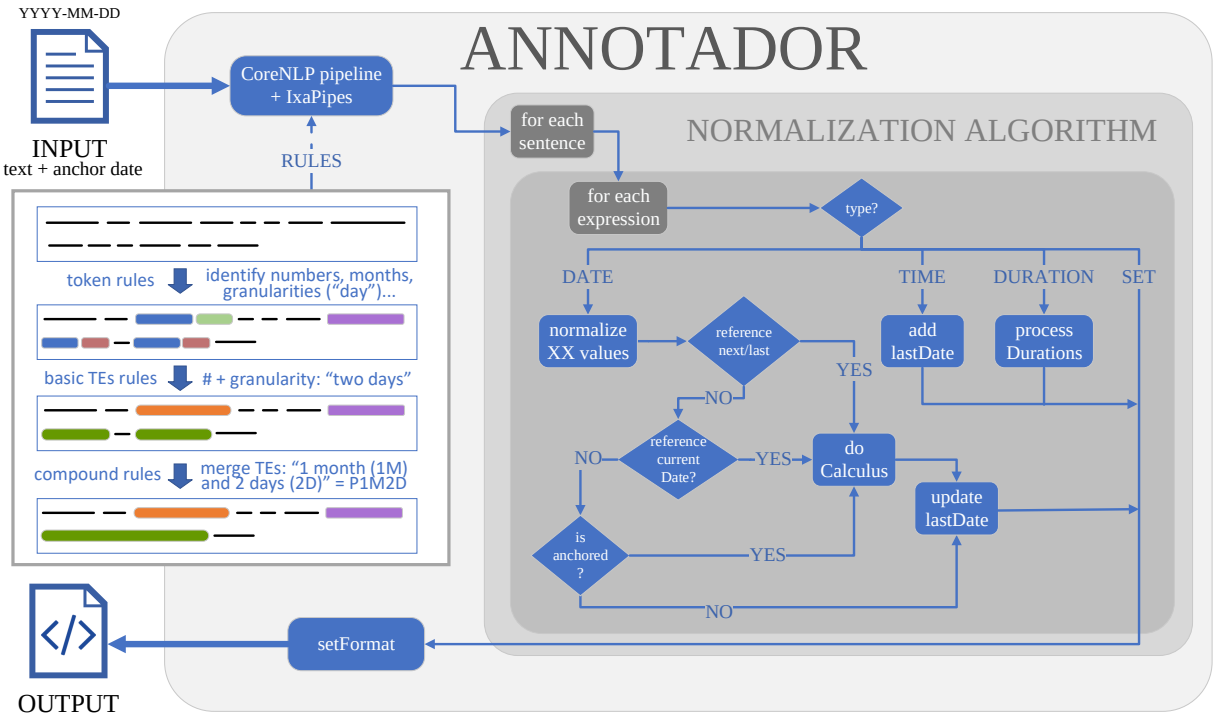

Fig. 1. Pipeline of Annotador. First we process the text using the CoreNLP pipeline (where we added IxaPipes' models). In the TokensRegex annotator we use our customized rules, that process the input text in different stages. Then these expressions are passed to the normalization algorithm, that process them defferenty depending on their type. In the case of DATEs, we first normalize the unknown values, marked with X (e.g., "XXXX-05" means we know it is the month of May but we have no further info on the year, so we normalize it to the anchor date). Then we can have references to next or last points in the calendar (e.g., "last December"), references to the current date at some specific granularity ("this month") or anchorings where we have to add or subtract specific amounts of time (e.g. "one year and two months ago"). Once we finish with this processing, we store the last date (to know the value of possible anaphoras in the same sentence) and we go for the next expression. We do this for all the sentences in the text (at each sentence restarts the last date variable, used as anchor date within the sentence, to the original anchor date if different), and then we return the annotated text in the required format (for now, TimeML TIMEX3 tags or JSON).

erwise it will be empty[11]) and finally (4) the last rule applied, so the reasoning that produced the result can be traced. In the following subsections we will detail how the rules are applied and how we generate these results via the intermediate annotation tags.

3.1.1. Basic tokens

In a first stage, Annotador detects basic token-based relevant expressions, such as:

- Numerals: numerals, either with numbers of words. This is a non-trivial task, as standard NER systems and POS taggers do not recognized numerals when represented with words, as in the example "mil cuatrocientos noventa y dos" ("one thousand four hundred and ninety-two").
- Names of months, days of the week and seasons: here it was necessary to check the POS tagging, since some of them, such as "abril" ("April"), "julio" ("July") or "domingo" ("Sunday"), are also personal names in Spanish.
- Granularities: we distinguish here between DGRANULARITY (anything bigger than a day, included, and that is considered DATE by the standard) and TGRANULARITY (anything smaller than a day, considered TIME by the standard), but for instance in the case of DURATIONs (like "two days", "one hour") they share common rules. Regarding the calculus, each granularity has information associated. For instance, we know that centuries ("siglos" or "centurias" in Spanish) are measured in years, and that each one corresponds to 100 years. The concept century (stored as "100_YEARS" in the system) has therefore an associated granularity of years ("Y" regarding the

[11] *freq* denotes the frequency a SET expression is repeated (e.g., one month for the expression "monthly").

standard) and an associated amount of 100, and it is identified when the lemmas "siglo" or "centuria" are used. If eventually we wanted to use another synonym, we would just have to add in the list of granularities of our rule file a new entry that maps to "100_YEARS".
- Parts of the day or specific relative days: such as "tarde" ("afternoon") or "ayer" ("yesterday").

Annotador also detects other expressions, such as ordinals and roman numerals, and assigns them a value. All of them are tagged with basic annotations, such as numeric values and the type of expression, that will be used afterwards in other rules.

3.1.2. Basic temporal expressions

Once the most basic expressions are identified, the next task is to combine them to detect temporal expressions. Some rules we can find at this stage are shown in Table 2, where we can see how the result of some rules rely on previous annotations by other rules.

Apart from the results shown in the table, also other internal tags are stored, depending of the type of temporal expression:

- TIME: in the case of TIME, Annotador stores the hour, the minute, the second and the part of the day of the temporal expresion.
- DATE: for DATEs, Annotador keeps the day, the week, the day of the week, the month and the year of the temporal expresion.
- DURATION: for DURATIONs, Annotador stores the granularities in the temporal expresion, such as the amount of years, days and hours.

Although in most common temporal expressions these values are never used, sometimes we will find time expressions where part of the info is omitted in its extent (e.g. in "from one to two days", the first expression includes no granularity). Being able to retrieve it from close expressions will be useful. How we do this kind of processing is explained in the next subsection.

3.1.3. Compound expressions

These rules mainly target time expressions where some information from one time expression must be used in another for normalization (e.g., the previously mentioned "from one to two days"). To this aim we will use the information from the rules presented in the previous subsections. Some examples of these types of rules can be found in Table 3.

3.1.4. Literal expressions

Apart from the previous rules, there are some token-based rules that target literal expressions, such as bank holidays or specific noun phrases. Some of these expressions are shown in Table 4. Please note that some of the expressions (such as #2 "el ayer" and #3 "el día de mañana") include time expressions with a different meaning ("ayer" means "yesterday", and "mañana", "tomorrow" or "morning"), so it is necessary to capture these expressions literally and avoid conflicts to the alternative interpretations. In fact, the word "mañana" is specially tricky, since it produces several expressions in Spanish, summarized below:

- "mañana" (femenine noun) means "morning".
- "mañana" (adv) means "tomorrow".
- "pasado mañana" (adv) means "the day after tomorrow".
- "pasado" (adv) has the same meaning as "pasado mañana" (this is, "the day after tomorrow").
- "pasado" (noun or adjective) means "past" (noun or adjective).

Additionally, idioms and structures containing the term "mañana" change depending on the language variant. The expression "in the morning" is said with the Spanish of Spain "por la mañana", but "en la mañana" in other varieties of Spanish used in Latin America. Of course, most POS taggers have a lot of problems to correctly annotate these words in their variate senses. We therefore have several rules just to disambiguate them in order to maximize accuracy when dealing with these polysemic expressions.

3.2. Normalization Algorithm

Once the rules are applied, the results are processed through a normalization algorithm that decides the final value of each expression.

For calendar calculus, we rely on the widely-used library JodaTime[12], that supports basic operations such as adding months or days to a date, or converting dates from different formats. Nevertheless, for more complicated operations, such as finding out the calendar date of a specific weekday (e.g., "the next Thursday" depends on the day of the week we are) or working with seasons, we developed a set of specific functions that complement JodaTime utilities.

[12] `https://www.joda.org/joda-time/`

Table 2
Examples of rules used to detect basic temporal expressions.

#	Example	Pseudo-pattern	Tagged as	Value in the example
1	dos días two days	number + granularity	P + value number + value granularity	DURATION (P2D)
2	cada dos semanas every two weeks	cada + [DURATION]+	value of DURATION	SET (P2W)
3	a las tres at three	a? + las + [hour] + ![noun]	"T" + hour + ":00"	TIME (T3:00)
4	las tres menos 5 five to three	a? + las + [hour] + menos + [minutes]	"T" + (hour-1) + ":" + (60-minutes)	TIME (T2:55)
5	las tres de la tarde three in the afternoon	[TIME] [de|en] + [la|el] + [PARTDAY]	am/pm value of TIME	TIME (T15:00)
6	el 1 de Mayo de 1991 May 1, 1991	[day] + de + [month] + de + [year]	year + "-" + month "-" + day	DATE (1991-05-01)

Table 3
Examples of compound rules.

#	Example	Pseudo-pattern	Tagged as	Value in the example
1	dos meses, tres semanas y un día two months, three weeks and one day	[[DURATION]+ [,|y]]+ [DURATION]+	P+value of each duration	DURATION (P2M3W1D)
2	el 1, el 2 y el 3 de mayo de 2011 1st, 2nd and 3rd May 2011	[dayNum [,|y]]+ [DATE]	each dayNum gets the info from DATE	DATE (2011-05-01, 2011-05-02, 2011-05-03)
3	mayo y junio de 2060 May and June of 2060	[Month [,|y]]+ [DATE]	each Month takes the year from the DATE	DATE (2060-05, 2060-06)
4	de uno a dos años from one to two years	[de|entre...] num [a|hasta...] [DURATION]	num inherits granularity from DURATION	DURATION (P1Y, P2Y)

Table 4
Some literal expressions.

#	Spanish Expressions	English Synonyms	Tagged as
1	hoy en día, a día de hoy, en la actualidad	nowadays, currently	DATE PRESENT_REF
2	previamente, antaño, recientemente, el ayer	previosuly, recently, the past	DATE PAST_REF
3	el día de mañana, en los próximos años	in the next years	DATE FUTURE_REF
4	Nochevieja, Fin de Año	New Year Eve	DATE XXXX-12-31
5	Halloween	Halloween	DATE XXXX-10-31

3.2.1. Normalization of DATEs

DATEs are undoubtedly the most tricky temporal type. Despite some rules can output directly its final value for absolute dates (e.g., "6 December 2019" will return "2019-12-06"), most of them will need further normalization. Here the concept of *anchor date* (the date we use as reference for calendar calculations) becomes crucial. At the beginning of the processing it will be the date provided to the system, but as the algorithm advances on the text, the last date found will be saved in order to use as reference date if needed. If for instance we had the text "El 4 de julio estudió por la mañana, pero no por la tarde." ("July 4, he studied in the morning, but not in the afternoon."), we understand that the mentions to parts of the day ("morning" and "afternoon") do not refer to the present day, but to the previously mentioned date "July 4". Annotador would therefore normalize it to "El 4 de julio (2019-07-04) estudió por la mañana (2019-07-04TMO), pero no por la tarde (2019-07-04TAF).".

When we detect a temporal expression (TE), we first check is if part of a DATEs is unknown (this is, the value returned by the rule includes "XXXX" or "XX"). If this is the case, we normalize it to the anchor date. This tends to happen when we have abstract mentions to days of the week or months (e.g, "in May" or "on Monday"). If this was not the case, there are two options: the value was absolute (this is, it is already the final value), or it is anchored. If it is the latter, we can

find several types of anchoring, that we will analyze hereunder:

1. The TE refers to a previous or a future specific date, day of the week, weekend, month or season.
2. The TE refers to a specific granularity of the anchor date (e.g. "this month").
3. The TE refers to a point in time resulting from adding or subtracting some duration to the anchor date (e.g. "yesterday" means to subtract one day to the present day).

The first case comprehends expressions such as "next friday" or "last summer", where the anchor date must be taken as reference to decide to which date they refer to. These expressions should not be confused with others such "last month". While this expression consist just on subtracting an amount of time to the anchor date (one month in the example), the ones we target in this section require a bit more sophisticated normalization. If we say "last summer" in December 1991, we make reference to the summer of 1991, but also if we say it in March 1992. But in May 1991, we would refer probably to 1990. The same happens with days of the week, weekends, specific dates (e.g. "last 5th March") or literal months (e.g. "next October"). To deal with these expressions, we created a set of functions that work over JodaTime on each specific granularity.

The second case focuses on expressions such as "this month", "that day" or "the year", where the time expression refers to some granularity of the anchor date. It is not always a value that can be directly extracted from the anchor date (such as the day, the month or the year), since it can also be a bigger granularity, such as the century the current date belongs to (e.g., "this century"). The rules of the system in this case return the desired granularity, and the normalization algorithm infers the correct normalized value from the anchor date.

Last case of anchoring implies adding or subtracting durations, such as in the expressions "yesterday" and "the day after tomorrow"[13]. In this case, our rules gather all the DURATIONs and express them as a concatenation (e.g. "P3M2W1D" for "three months (M), two weeks (W) and a day (D) ago"). Then the algorithm iteratively uses JodaTime and our functions to add or subtract each of them. So if our anchor case in the previous example was "2019-12-20", the system would first subtract three months (2019-09-20), then two weeks (2019-09-06) and finally one day (2019-09-05), obtaining the desired date. The part of the algorithm doing these operations is disabled in the current version of Annotador for expressions such as "two days ago" or "in three months and two weeks" –working just for expressions like "tomorrow"– because of the TimeML guidelines, that specifically asks to annotate them as DURATIONS – but it can be reactivated if required, since it is useful for tasks such as timeline creation.

Complementary expressions to this last case would be for instance "the rest of the year" ("lo que resta/queda de año") or "the part of the month that already passed" ("lo que va/llevamos de mes"), where we return a composed duration (e.g., for the 2nd March we would return "P2M1D", two months and a day).

3.2.2. Normalization of other types of TE

Not only DATEs are normalized by our algorithm. The final value of DURATIONs is also an output of the normalization algorithm. This process similar to the parsing of DURATIONs introduced in the last case of the previous Section 3.2.1.

Also TIME expressions are normalized by this algorithm. If for instance we found a part of a day (e.g. "night", normalized by the rules as "TNI") or a time (e.g. "at 7 pm"), the system would anchor it to the current anchor date (e.g. "2019-12-20TNI" and "2019-12-20T19:00", respectively).

3.3. Availability

Annotador is available as a repository in GitHub[14], where the code and the rules can be downloaded and modified under a GNU GPL-3.0 license[15]. The code also includes methods to test the tool against different corpora. It requires no external installation besides Maven dependencies.

There is also a visual demo[16] where the users can test the tool[17]. The service can also be invoked via cURL or Postman (a Postman collection of requests is also available to facilitate its use), receiving plain text and returning the annotations in TIMEX3 or JSON.

[13] As explained in Section 3.1.4, "pasado" or "pasado mañana" is a particular expression in Spanish denoting "the day after tomorrow".

[14] The code is usually available in GitHub, but the repository is currently private for the sake of the double blind review process. It can be sent to the reviewers if required.

[15] `https://www.gnu.org/licenses/gpl-3.0.html`

[16] Anonymized here: `http://lke42-243314.appspot.com/`

[17] This demo is a basic version of the original one, prepared for the review, since the main one cannot be shown due to anonymization.

4. HourGlass Corpus

4.1. Inception

During the development of Annotador, the scarcity of Spanish corpora became evident. We also noticed that available corpora usually do not cover all the possible time expression variance and that they do not facilitate detecting flaws in temporal taggers. Additionally, when we add new rules to a temporal tagger (or we retrain it in the case of machine-learning-based ones), we risk to stop correctly covering expressions we did correctly before the changes. Also, Spanish corpora just include news and historical Spanish texts, leaving aside many time expressions (such as colloquial ones) that do not tend to appear in those kinds of texts. Finally, these corpora just cover Castillian Spanish, leaving aside other Spanish-speaking countries. For all these reasons, we decided to build a corpus to facilitate systematically testing temporal taggers in Spanish, including expressions from different registers and countries. We can divide this corpus in two different parts.

The first part of the corpus (to which we will refer to as *synthetic*) is a set of short texts mainly developed for testing Annotador. This does not mean that Annotador covers them all, but that the expressions were written specifically for testing a temporal tagger, taking into account the temporal tagging task and the TimeML standard.

The second part of the corpus (the *people* part) was built by contributors foreign to the temporal tagging task, and mostly to NLP. Twelve professionals from several disciplines, ages and Spanish-speaking countries provided time expressions for this part of the corpus. They were asked to add them in a Google spreadsheet or just tell the authors using any means of communication during a period of two weeks. Therefore, some of these sentences were thought specifically for the corpus, but others are sentences they used during real conversations and chats and that they also asked us to include because they considered them to contain temporal expressions. Our volunteers were given a basic call for expressions including some examples of the four types of time expressions in the TimeML standard. During this process, we found out that while volunteers find more or less intuitive other tasks in NLP (for instance, what is a *named entity* is usually more or less clear to people out of the domain), they find difficult to distinguish what is a temporal expression and what is not. Most of the expressions we got from our contributors are not envisaged in the standard, and should probably me marked as SIGNALS (the tag in TimeML used to mark expressions with some temporal information but that are not time expressions per se) or temporal relations. For this reason, some texts in this part of the corpus have no annotations. Other texts were too ambiguous to be annotated following the standard, and were therefore not added to the corpus; nevertheless, we decided to make them available to foster discussion.

4.2. Pre-processing of the corpus

In the final corpus we included 348 documents, 285 synthetic texts and 63 from our contributors – more information can be found in Table 5. We also have four additional texts from our contributors whose annotation was ambiguous, so we did not include them in this final corpus (although we made them available along with the corpus).

For annotation, we first checked the sentences, specially the ones by contributors, in order to confirm that they could be tagged following the standard – the ambiguous expressions were marked and left aside. Also, we added comments for some of them. Then, we formalized the tags of each part of the corpus.

In the case of the *synthetic* part, we normalized the different comments of each text (such as "to check if this way of expressing dates is covered" or "should not be tagged") into a series of more than twenty normalized tags in order facilitate testing. Some of these tags are for instance *Dates*, *Fractions* or *False* (referring to a *false positives*, such as the expression "50/2/1991", that should not be tagged despite of looking like a date). These tags are specially useful in case we update a temporal tagger and we want to check if our coverage of certain time expressions changed, and also if we are interested just in some type of expressions.

For the *people* part, we normalized the tags differently, including for instance the tag *standard* if the sentence would be covered by the TimeML standard, *yes* if it should be tagged but it is not clear how following the standard TimeML, *no* if it is a expression involving "temporal words" but where their meaning changes (or if it should not be tagged despite the contributors of the corpus thought it included some time expression), and *special* if it has some special meaning –an analysis of some examples of time expressions tagged as *special* can be found in Section 6. Besides the tags, we also added the register of each sentence. Among this register we find *colloquial*, *chat* (these sentences were

Table 5
Statistics on the HourGlass corpus, overall and for each part. They are given as total (e.g. the amount of tokens in the whole *synthetic* corpus) and on average (average of tokens per document).

	synthetic		people		all	
	total	avg	total	avg	total	avg
documents	285		63		348	
sentences	292	1.03	67	1.06	359	1.03
TIMEX3	341	1.20	58	0.92	399	1.15
DATE	165	0.58	25	0.40	190	0.55
DURATION	102	0.36	21	0.34	123	0.35
SET	26	0.10	4	0.06	30	0.09
TIME	48	0.17	8	0.13	56	0.16
tokens	1927	6.76	688	10.92	2615	7.51
adjectives	69	0.24	20	0.32	89	0.26
adverbs	69	0.24	35	0.56	104	0.30
nouns	332	1.17	125	1.99	457	1.31
NPs	25	0.09	16	0.25	41	0.12
verbs	155	0.54	112	1.78	267	0.77

extracted from chats, so grammar is less strict), *latin expressions* (very common in legal texts), *Latin American expressions*, *phrases* and one *literary* sentence.

4.3. Annotation of the corpus

Once the tags were added, we started the annotation process. To facilitate this task, we first used a temporal tagger on the texts and we then manually corrected its annotations and added missing ones. In order to avoid bias, we did not use our tagger to this task, but HeidelTime. Then, we performed a first round of annotations based on the guidelines available for Spanish [1]. Afterwards, a second round of annotations was done, reviewing the previous ones and correcting them when needed. The same anchor date ("2019-12-20") was used for the annotation of all the documents. The statistics of the corpus are detailed in Table 5.

The corpus is published (under a GNU GPL-3.0 license) as plain texts without annotations, TimeML files without annotations and TimeML files with annotations[18], along with excel files including the metadata of each document (namely, id, content, annotated text, if it belongs to the *synthetic* part or the *people* part, tag, register and comments on the annotations)[19].

[18] After the anonymization period we will publish it in Zenodo.

[19] To facilitate testing just one part of the corpus, files are named with their id, five numbers. If the first number is a 0, the file belongs to the *synthetic* part, while a 9 means it is from the *people* part).

The corpus and additional information about it can be found in its website[20], along with the result of the different taggers tested on it.

5. Evaluation

In this section we will present our results against HeidelTime and SUTime[21]. We have done the evaluation in terms of *precision* (this is, the share of hits among the expressions tagged by the tools), *recall* (the share of hits among all the expressions to be tagged) and *F1-measure* (the average of *precision* and *recall*). We will consider this metrics *lenient* (this is, we consider a hit a partially tagged expression, even if not all its extent is marked by the tagger), *strict* (just expressions tagged exactly as in the test are considered correct) and *average* (average of *lenient* and *strict*). In order to extract these metrics from the results of the taggers, we used the software GATE [25]. We considered for evaluation the same attributes as in the TempEval 2 challenge, (1) the identification the extent of the TE (*extent*), (2) the identification of the type of TE (*type*) and (3) the normalization (*value*). Apart from our HourGlass corpus, we also used TempEval 2 corpus for evaluation[22].

5.1. HourGlass corpus

Table 5.2 shows the result of the taggers on the HourGlass corpus. Despite Annotador gets the best

[20] `http://www.lke42-243314.appspot.com/hourglass`

[21] HeidelTime was called using the following parameters: "News" as type of text, "Spanish" as language and "TreeTagger" as POS tagger. SUTime was invoked directly, not via the NER Annotator, as in the example code available in its documentation (`https://nlp.stanford.edu/software/sutime.shtml`), but using the Spanish properties. Although we also tried to evaluate the temporal tagger TIPSem running it on different machines and configurations, we were never able to use it to process Spanish texts (despite we succeeded for English) due to the unavailability of some auxiliary software required by TIPSem. Nevertheless, we would want to thank its creator for his support and help during the process.

[22] As we explained in Section 2, Spanish corpora is really scarce: TempEval 3 test dataset is not available, TimeBank ModeS is for old Spanish, TimeBank has the same documents as TempEval 2 and 3 and MEANTIME corpus does not annotate all the time expressions in a document, so it was not suitable. Also, it must be noted that the scorer available for TempEval 2 did not include the key documents for Spanish, and did not work. We therefore got the key documents from GitHub (`https://github.com/AntonFagerberg/Temporal-Information-Extraction/tree/master/tempeval2-data`) and used GATE for comparing the results.

results, all the taggers shared some common errors. For instance, none of them found the colloquial expression "en cero coma", that means "in seconds" (doc 90001). In doc 90065, not HeidelTime nor SUTime found the expression "lo vuestro dura 1h, no?" ("your stuff lasts 1h, right?"; Annotador correctly marked it, but wrongly considered it a TIME instead of a DURATION. Similarly, in the case of compound durations (such as "1 año, 6 meses y un día", "1 year, 6 months and one day", from doc 00060), each tagger performed differently: Annotador correctly marked it all as a full expression, HeidelTime tagged each part individually and SUTime did not recognize any time expression. Finally, both HeidelTime and SUTime have problems when recognizing literal numbers in Spanish – an example of this can be seen in doc 00011, where in "In the year one thousand." HeidelTime just recognized "one year" and SUTime tagged nothing. This problem also appears when dealing with polysemic expressions such as "pasado" and "mañana" (previously introduced in Section 3.1.4); doc 00008 includes the text "Ya lo veremos pasado mañana." ("We will see it the day after tomorrow."), where SUTime just recognizes "mañana" as "tomorrow" and HeidelTime tags the expression correctly but considers that it refers to the morning of the previous day. Regarding Latin American Spanish, only Annotador recognizes expressions like "Cinco para las 11." ("Five to eleven.", doc 90053). The output of the different taggers can be found in the website of the corpus.

5.2. TempEval 2

In the TempEval 2 corpus, we had 175 documents for train and 35 documents for test. Since the documents were in the .tab format but the Python scorer facilitated in the website did not work, we transformed these tab files into plain text documents for testing the output of the temporal taggers using GATE.

In Table 7 we show the results obtained by Annotador, HeidelTime and SUTime. As in the previous evaluation, SUTime precission is generally its highest metric, although it is in most cases beaten by Annotator and HeidelTime. On the other hand, Annotator's recall is the highest in all cases. Regarding F1-measure, Annotador tends to be better for detecting the *extent* of the tag and its *type*, while HeidelTime is slightly better on normalizing the *value*. Overall, Annotador is better than HeidelTime in most of the metrics, having similar results when not, and both tend to surpass SUTime.

6. Challenges

Despite of the good numbers of the temporal taggers, there still are open issues. *Context-free TE* refer to fixed instants or intervals of time irrespective of any other consideration.

Context-dependent TE (CDTEs) refer to precise instants or intervals, but in order to determine them, some additional context information is necessary. This context information can be present in the text in one form or another, from very explicit mentions to indirect hints from which it can be inferred. In the worst case, the context information will be tacit knowledge, shared only between the writer and a specific reader or reader type. We identify here different types of CDTEs, where context information is necessary to determine (i) whether a group of words is a TE or (ii) how to normalize the TEs.

TE dependent on temporal information Whereas Annotador considers the anchor date as a date of reference to resolve relative references (e.g. "tomorrow"), the temporal information to be considered to disambiguate can be more complex. In the sentence: "Entre el golpe de estado del 18 de brumario y el 3 de nivoso." ("Between coup of 18 Brumaire and 3 Nivôse."), "nivoso" has two senses that need to be disambiguated by temporally framing the text. Besides the French Republican calendar, also other calendars have named historical facts, such as the Julian calendar and the October Revolution, that actually happened in November according to our calendar. Additionally, some countries have their own calendars and date elements, such as the Japanse era system, the Chinese lunisolar calendar or the Persian Solar Hijri calendar.

TE dependent on geographical information Geographical information can refer to physical geography or to political geography information. An example of the former is *spring*, which depends on the hemisphere, and an example of the latter is *Día del Niño* (*International Children's Day*), which depends on a political decision different for every jurisdiction –it is for instance celebrated the 15th of April in Spain, but the 30th in Mexico. Geographical information may also help to correctly normalize certain date formats, since 09/10/2019 means 9th October in Europe but 10th September in the United States. Finally, dialects also imply different ways to refer to time, such as the Latin American expression "cinco para la una" ("five to one"), that is not used in Spain (where it is usually expressed as "la una menos cinco", "one minus five").

Table 6

Results of the temporal taggers in the HourGlass corpus. Annotador has the highest results, although HeidelTime also shows good performance. All the taggers show worse performance in comparison to the TempEval 2 corpus, although the difference is smaller in the case of Annotador.

		strict			lenient			average		
Tagger	Attribute	P	R	F1	P	R	F1	P	R	F1
Annotador	value	0.7231	0.7068	0.7148	0.7949	0.7769	0.7858	0.7590	0.7419	0.7503
	type	0.7923	0.7744	0.7833	0.8846	0.8647	0.8745	0.8385	0.8195	0.8289
	extent	0.8333	0.8145	0.8238	0.9462	0.9248	0.9354	0.8897	0.8697	0.8796
Heidel	value	0.5672	0.4762	0.5177	0.6358	0.5338	0.5804	0.6015	0.5050	0.5490
	type	0.6060	0.5088	0.5531	0.8239	0.6917	0.7520	0.7149	0.6003	0.6526
	extent	0.6239	0.5238	0.5695	0.8716	0.7318	0.7956	0.7478	0.6278	0.6826
SUTime	value	0.3019	0.0802	0.1267	0.4528	0.1203	0.1901	0.3774	0.1003	0.1584
	type	0.4717	0.1253	0.1980	0.8019	0.2130	0.3366	0.6368	0.1692	0.2673
	extent	0.4717	0.1253	0.1980	0.8868	0.2356	0.3723	0.6792	0.1805	0.2851

Table 7

Results of the temporal taggers in the TempEval 2 corpus –best metrics for each category are highlighted in bold. Although HeidelTime is slightly better at finding the normalized value (0.0027 on average), Annotador is better in the rest of metrics. SUTime rules obtain on the other side high precision but low recall.

		strict			lenient			average		
Tagger	Attribute	P	R	F1	P	R	F1	P	R	F1
Annotador	value	0.8021	**0.7778**	0.7897	0.8281	**0.8030**	0.8154	0.8151	**0.7904**	0.8026
	type	0.8438	**0.8182**	**0.8308**	0.9063	**0.8788**	**0.8923**	**0.8750**	**0.8485**	**0.8615**
	extent	0.8646	**0.8384**	0.8513	0.9323	**0.9040**	**0.9179**	0.8984	**0.8712**	**0.8846**
Heidel	value	**0.8418**	0.7525	**0.7947**	**0.8644**	0.7727	**0.8160**	**0.8531**	0.7626	**0.8053**
	type	**0.8531**	0.7626	0.8053	0.8870	0.7929	0.8373	0.8701	0.7778	0.8213
	extent	**0.9040**	0.8081	**0.8533**	0.9435	0.8434	0.8907	**0.9237**	0.8258	0.8720
SUTime	value	0.6377	0.2222	0.3296	0.8261	0.2879	0.4270	0.7319	0.2551	0.3783
	type	0.6522	0.2273	0.3371	**0.9275**	0.3232	0.4794	0.7899	0.2753	0.4082
	extent	0.6667	0.2323	0.3446	**0.9565**	0.3333	0.4944	0.8116	0.2828	0.4195

TE dependent on the register The jargon can also affect to TE identification and normalization. There are a lot of expressions in Spanish where non-temporal words are used in a temporal sense, some of them included in the HourGlass corpus. Examples of these are the expressions "Él tiene 37 castañas" ("He has 37 chestnuts") and "Él tiene 37 tacos" ("He has 37 tacos"), both meaning "He is 37 years old". Other expressions can also change their meaning in a meronymic way, such as is the case of "Tiene 30 primaveras/abriles ya" ("He already has 30 springs/Aprils"), where a part of the year (the spring or the month of April) represents the whole year. Similarly, we also have many idioms involving temporal expressions that should not be tagged, such as the phrases "Hasta el 40 de mayo no te quites el sayo" ("Until 40th May, do not take off the jacket", meaning that the beginning of June can still be chilly), "En abril, aguas mil" ("In April, thousand waters", meaning that April is usually rainy) or "A buenas horas mangas verdes" ("At good hours, green sleeves", meaning someone acted too late), and expressions like "en el último minuto" ("in the last minute", meaning close to a deadline), where "minute" should not either be be tagged. Regarding Latin American Spanish, there also exist a lot of similar idioms and expressions, such as "la hora del moro" ("the hour of the moor"), that means "lunch time" in the Dominican Republic.

Despite the problem of resolving CDTEs has already been partially studied [26], to the best of the authors' knowledge there are no full-working solutions.

7. Conclusions

In this paper we introduced Annotador, a temporal tagger for Spanish texts also able to process English texts. We also identified the lack of Spanish cor-

pora to test this kind of tools. We created to this aim the HourGlass corpus, where we tagged and annotated both syntethic texts and expressions from contributors unrelated to temporal tagging task and with different backgrounds. This corpus therefore contains variate time expressions from different Spanish-countries and linguistic registers, but also common expressions in Spanish involving some *temporal words* (such as names of months) that should not be tagged or should be tagged differently than in the literal sense. We used it, together with the TempEval 2 dataset, to evaluate Annotador and compare its results to the performance of two state-of-the-art taggers, HeidelTime and SUTime –Annotador surpassed them in both cases for most of the metrics, maintaining similar results in both corpora. Finally, we also analyzed the particularities of the Spanish language and the challenges of context-dependent time expressions.

The work presented herein is a prerequisite for future work which focuses on the automatic extraction of events and timelines, mainly from legal texts in Spanish and English. Regarding time expression extraction and normalization, we also want to extend Annotador, implementing interval identification and improving normalization by using the dependencies and the verbal tenses in a sentence (e.g., when we say "on Friday" we normalize it to the Friday from the current week, but depending on the tense of the verb in the sentence we could confirm this or normalize to other surrounding Friday). We also want to work on the challenges and particularities identified for Spanish context-dependent expressions, that seems to be the line to cross by state-of-the-art temporal taggers.

Acknowledgement

This paper has been supported by the the project **ANONYMIZED FOR REVIEWERS**. We would also want to thank all the contributors to our corpus.

References

[1] R. Sauri *et al.*, "Annotating time expressions in Spanish TimeML annotation guidelines," 2010.

[2] J. Strötgen and M. Gertz, "Multilingual and cross-domain temporal tagging," *LREv*, vol. 47, no. 2, pp. 269–298, 2013.

[3] A. X. Chang and C. D. Manning, "Sutime: A library for recognizing and normalizing time expressions," in *LREC*, vol. 2012, pp. 3735–3740, 2012.

[4] J. Pustejovsky *et al.*, "TimeML: Robust Specification of Event and Temporal Expressions in Text," in *New Directions in Question Answering*, pp. 28–34, 2003.

[5] J. Strötgen *et al.*, "Temporal tagging on different domains: Challenges, strategies, and gold standards.," in *LREC*, vol. 12, pp. 3746–3753, 2012.

[6] X. Zhong *et al.*, "Time expression analysis and recognition using syntactic token types and general heuristic rules," in *Proceedings of the 55th Annual Meeting of the ACL*, vol. 1, pp. 420–429, 2017.

[7] S. Bethard, "ClearTK-TimeML: A minimalist approach to TempEval 2013," in *Proceedings of the Workshop SemEval 2013*, pp. 10–14, ACL, June 2013.

[8] L. Derczynski *et al.*, "Usfd2: Annotating temporal expresions and tlinks for tempeval-2," in *Proceedings of the Workshop SemEval*, pp. 337–340, ACL, 2010.

[9] K. Lee *et al.*, "Context-dependent semantic parsing for time expressions," in *Proceedings of the 52nd Annual Meeting of the ACL*, vol. 1, pp. 1437–1447, 2014.

[10] M. Verhagen *et al.*, "Automating Temporal Annotation with TARSQI," in *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, pp. 81–84, ACL, 2005.

[11] I. Mani *et al.*, "Robust temporal processing of news," in *Proceedings of the 38th annual meeting on ACL*, pp. 69–76, 2000.

[12] N. Chambers *et al.*, "Dense event ordering with a multi-pass architecture," *Transactions of ACL*, vol. 2, pp. 273–284, 2014.

[13] H. Llorens *et al.*, "Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2," in *Proceedings of the Workshop SemEval*, pp. 284–291, ACL, 2010.

[14] A. X. Chang *et al.*, "TokensRegex: Defining cascaded regular expressions over tokens," Tech. Rep. CSTR 2014-02, Department of Computer Science, Stanford University, 2014.

[15] C. D. Manning *et al.*, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proceedings of the 52nd Annual Meeting of the ACL, System Demonstrations*, pp. 55–60, 2014.

[16] L. Ferro *et al.*, "Tides temporal annotation guidelines-version 1.0.2," tech. rep., MITRE Corporation, 2001.

[17] W. Styler IV *et al.*, "Temporal annotation in the clinical domain," *Transactions of ACL*, vol. 2, pp. 143–154, 2014.

[18] R. Sprugnoli and S. Tonelli, "One, no one and one hundred thousand events: Defining and processing events in an inter-disciplinary perspective," *Natural Language Engineering*, vol. 23, no. 4, p. 485–506, 2017.

[19] J. Pustejovsky *et al.*, "The timebank corpus," in *Corpus linguistics*, vol. 2003, p. 40, Lancaster, UK, 2003.

[20] A.-L. Minard *et al.*, "Meantime, the newsreader multilingual event and time corpus," in *Proceedings of LREC 2016*, 2016.

[21] P. Mazur *et al.*, "Wikiwars: A new corpus for research on temporal expressions," in *Proceedings of EMNLP 2010*, pp. 913–922, ACL, 2010.

[22] J. Tabassum *et al.*, "Tweetime: A minimally supervised method for recognizing and normalizing time expressions in twitter," *arXiv preprint arXiv:1608.02904*, 2016.

[23] A. X. Chang and C. D. Manning, "Tokensregex: Defining cascaded regular expressions over tokens," *Tech. Rep. CSTR 2014-02*, 2014.

[24] R. Agerri *et al.*, "Ixa pipeline: Efficient and ready to use multilingual nlp tools.," in *LREC*, vol. 2014, pp. 3823–3828, 2014.

[25] H. Cunningham *et al.*, "Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics," *PLOS Computational Biology*, vol. 9, pp. 1–16, 02 2013.

[26] K. Lee *et al.*, "Context-dependent semantic parsing for time expressions," in *Proceedings of the 52nd Annual Meeting of the ACL*, vol. 1, pp. 1437–1447, 2014.