# Temporal taggings of Entities in Text

**Haozhou Pang**
Department of Computing Science
University of Alberta
haozhou@ualberta.ca

**Yourui Guo**
Department of Computing Science
University of Alberta
yourui@ualberta.ca

## Abstract

The problem addressed in this paper aims at searching for temporal taggings that are significant to entities mentioned in articles. This task mainly consists of three phases: first, name entity recognition, which seeks phrases or strings that refer to named entities like organizations, person name and places and so on. Second, temporal expression detection, which is the process of looking for expressions in text that refers to time points or time intervals. Third, the mapping from entities to temporal expressions. In this project, we proposed a model that utilizes the lexical and syntactical information in text to tackle the problem of temporal tagging.

## 1 Introduction

Entities in text can be related to some time points or intervals mentioned in an article, and the demand of knowing temporal expressions that are significant to entities in articles is increasing over the years. For example, utilizing the temporal expressions can help us to look for the named entity it truly refers to. When we read a newspaper about Bush, we can interpret the named entity as George W. Bush knowing his birthdate being July 6, 1946 mentioned in the article. Also, extracting temporal expressions related to entities helps us to construct knowledge bases. As we know that most of knowledge bases contain a small quantity of temporal taggings for named entities. Thus, the content of existing knowledge bases can be expanded by providing temporal information that was taken from any source of text.

Temporal tagging detection is the process of looking for expressions in text that refers to time points or time intervals. Few works have studied on detecting temporal taggings (Chang and Manning, 2012), (Strötgen and Gertz, 2012). SUTime (Chang and Manning, 2012) is a library for recognizing and normalizing time expressions. It transforms time expressions like October 1963 to the normalized value of 1963-10 and type of DATE. Normalizing temporal expressions reduces the chance of mis-recognizing the same date value to different ones, and the result of detecting the pairs of entity and temporal tagging can be more accurate as well.

Named entities recognition seeks phrases or strings that refer to named entities like organizations, person name and places and so on. Natural Language ToolKit (NLTK) (Bird et al., 2009) is a platform that provides us with interfaces to work with human natural language data. NLTK is utilized for natural language processing such as named entity recognition. The basic idea of recognizing named entities is to consider the task as a noun-phrase chunking. Raw text will be split into sentences first, and each sentences will be chunked by word tokenizer. Then, next step is to seeking noun phrases among chunks by tagging each chunk with pos-tagger. The last step is to use relation detection to search for potential related entities to ensure the detected entity is meaningful.

The problem addressed in this paper aims at

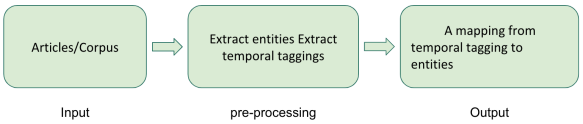


Figure 1: The visualization of the process

searching for temporal taggings that are significant to entities mentioned in articles. The input can be a corpus containing multiple raw texts such as news articles, web data and social media. The output is a list of pairs that consist of both entities and temporal taggings where for each pair the entity is highly related to the temporal tagging. In this work, we need to find out all temporal taggings mentioned in plain text, and also look for entities related to them. Then, we should sort out the list of result by ranking the relatedness of entity and temporal tagging in each pair among all the pairs, and retrieve top k result with highest relatedness.

As shown in figure.1, our task could be formulated in such way: given a set of articles with entities and meaningful temporal expressions, since we are dealing with raw text data of any format and any length, we first use the nltk sentence tokenizer (Bird et al., 2009) to split the text into sentences, then we preprocess the sentences by extracting the candidate entities and temporal expressions, we calculate the importance score by using a series of features for each pair of entity and temporal expression so that the output will be a list of ranked pairs. The importance score is measured by adding syntactic feature and lexical feature. In this way, we can extract the most important temporal information along with their corresponding entities from a article (plain text) without assistance of any external resources.

The remainder of the paper is organized as follows: Section 2 describes the related work that addresses the same problem using other methods. Section 3 introduces the dataset and methodologies used for detecting and ranking important pairs of entity and temporal tagging,. Section 4 describes the experiments and reports the results and analysis about out experiments. Finally comes the summary and conclusion about our contributions as well as directions for future work.

## 2 Related Work

There are some papers that studied on this problem, and the concept of knowledge base is addressed to solve it where individual entities and relationships among them are stored in a large repository. Knowledge base is widely used to store complex structured and unstructured information, and knowledge bases like YAGO (Suchanek et al., 2007), DBPedia (Auer et al., 2007) and freebase (Bollacker et al., 2008) are popular and public. The format of relation stored in knowledge base is a Subject-Predicate-Object (SPO) triple. Having the property that Knowledge base provides entities along with important facts about entities and other important temporal information, (Kuzey et al., 2016) proposed a method to detect temponyms and map a temporal tagging into a fact or event if present in knowledge base.

The entire process of detecting temponyms is divided into two parts. The first step is to extract important phrases from the free-text input and detect temponyms from extracted phrases. The second step is to disambiguate temponyms onto events stored in Knowledge base. Temponyms extracted from above methods consist of a large set of candidate entities. To prune candidates with low possibility of being mapped into events, mention-entity Dictionary is introduced to rank them. Top-k candidates are selected and a set of facts from Yago2 is selected if it contains the entity mentioned in top-k candidates. Some measures are applied to describe the relatedness for the mapping between temponym and facts.

However, the method that we proposed in this paper does not build on the same ideas proposed in this paper. Knowledge base like DBPedia has a small number of temporal taggings for each entity, but there might be many time points or intervals detected in raw text that we cannot retrieve from knowledge base. So, many of results extracted from plain text

will be lost as the knowledge base lacks of temporal information.

(Niu et al., 2012) proposed the system of Deep Dive to construct knowledge base. Unlike the construction of DBPedia (Auer et al., 2007) who extracts structured content from Wikipedia, Deep Dive makes the advantage of applying Natural Language Processing (NLP) to extract useful linguistic features from plain text. Deep Dive also performs web-scale statistical learning and inferences using data-management and optimization tools.

The basic working process of Deep Dive is to first extract useful relational features from input text, and then train the statistical model that presents the correlations between linguistic patterns and target relations with those extracted features. It then uses markov logic program to combine both statistical model and additional knowledge and transform relational features to relationships of entities. In this paper, both lexical and syntactic features are used to train statistical relation-extraction model. Lexical and syntactic features are both introduced in detail in the work proposed by Mintz et al. (Mintz et al., 2009), and we will also present both of them that we applied in this paper in section 3.

## 3 Method

The section presents information about how we approached the current task with more detailed description about the data set.

### 3.1 Data

Our dataset was constructed based on DBpedia dataset (version 2015-4). More specifically, we use the Extended Abstracts and Mapping-based Properties sections from DBPedia dataset. Extended Abstracts are used as the raw text input corpus, where the selected extended abstracts contain at least one temporal expressions. Mapping-based Properties are knowledge base that store relations among entities where the data is in the format of RDF. It provides important information of entities, and helps to construct the corpus that contains the most of useful temporal expressions needed for experiments.

We designed two test set for different purpose, the first dataset was constructed by extracting all the sentences that mention about $BarackObama$. This dataset contains 17148 sentences and 3491157 word tokens. The purpose of this dataset is to test whether our model can deal with the situation where one entity is related to multiple temporal expressions, and the experiment objective is to see how the different models rank the pairs differently.

The second dataset was constructed by extracting all the sentences that mention any one of the following name: $Donald\ Trump$, $Bill\ Clinton$, $Ronald\ Reagan$, or $Richard\ Nixon$. This dataset is more complicated since it contains far more name entities and meaningful temporal expressions. Also, the number of sentences / word tokens in the dataset are more than twice in the first dataset. The objective is to test whether our model can deal with larger corpus and more complex relations between entities and temporal expressions.

The detailed statistics of the datasets are summarized in table 1.

| | Dataset 1 | Dataset 2 |
|---|---|---|
| Num. of sentences | 17148 | 37656 |
| Num. of word tokens | 3491157 | 7184467 |
| Num. of entities | 36233 | 76318 |
| Num. of Temporal exp. | 17007 | 37988 |

Table 1: Statistics of the datasets

### 3.2 Baseline Algorithm

We first observed some articles that contains entities and meaningful temporal expressions, we find that one entity and its corresponding temporal expression are very likely to occur in the same sentences. For example, one sentence in our dataset is that: *In 2008, Barack Obama became the first African American to be elected president of the United States.*, the entity *Barack Obama* and the temporal expression *2008* appear in the same sentence,

and we find this hypothesis are very likely to hold in a well-written article. Therefore, our baseline algorithm uses this simple heuristic by assigning a score to such pair of entity and temporal expression. At the end, all the pairs will be ranked based on the frequency of the co-occurence in one sentence.

However, the main flaws of the baseline algorithm are that: first, the baseline does not catch the situation when the name entity and temporal expression are not in the same sentence. Second, the entities extracted by nltk are noisy, since we observed that some meaningless entities (*e.g D, St*) are also extracted by nltk, which results in a significant degrade of the performance since such meaningless entities will be involved in the output. In this project, we try to tackle these problems by introducing the lexical feature and syntactic feature, which will be discussed more in detail in next section.

### 3.3 Lexical Feature

In order to solve the situation that the entity and temporal expression are not in the same sentence, we introduce our lexical feature. It is formally defined as:

$$score = \sum(w/d)$$

where w is the weight of lexical feature, and d is the distance (number of sentences) from entity to temporal expressions. For example, one sentence in the dataset is that: *Barack Obama was elected over Republican John McCain and was inaugurated on January 20, 2009. Nine months later, he was named the 2009 Nobel Peace Prize laureate.* Then entity $BarackObama$ and the temporal expression $2009$ are not in the same sentence, but if we look at the context we will know that they are actually related. In this case, by applying the lexical feature, the pair <Barack Obama, 2009> will also receive partial weight.

### 3.4 Syntactic Feature

We try to tackle the problem of meaningless entities caused by nltk by introducing the syntactic feature. After observed some sentences in the dataset, we found that the entities usually appear at the subject position of a sentence. Therefore, we use the Stanford CoreNLP sentence parser (Manning et al., 2014) to extract the subject of a sentence. If the subject of a sentence happens to be an entity name, then we assign extra weight to such paris that contain the particular entity. In the example discussed in previous section, the sentence: In 2008, Barack Obama became the first African American to be elected president of the United States., the pair <Barack Obama, 2008> will be assigned a extra weight, since the entity Barack Obama is also the subject of the sentence.

Additionally, we take a sample of 10 sentences and we find that the entities appear in the 10 sentences are all proper nouns. The intuition is that the entities of a sentences are usually person names, locations, organizations, and etc. Syntactically, such entities typically have a pos-tagging of NNP (proper noun). Based on this observation, we use nltk pos-tagger (Bird et al., 2009) to extract all the proper nouns that appear in the dataset, and we extend our syntactic feature such that we assign extra weight to the pairs that the entity is a proper noun.

In this way, since the meaningless entities such as D are neither proper nouns nor subjects of a sentence. Assuming the weight given to syntactic feature is appropriate, we can filter out those meaningless entities.

### 3.5 Evaluation

The evaluation was conducted based on the top-50 results returned from each algorithm. We manually assess each result and report the precision estimated from the top-50 results.

## 4 Result

The section present the results of the current task followed by detailed error analysis.

### 4.1 dataset 1

Our baseline model works reasonably well in the first dataset, in the top 50 pairs, 18 of them

are correct. Our proposed model performs even better with 39 correct pairs out of 50.

| Entity | Temporal exp. |
|---|---|
| Obama | 2008 |
| Bar | 2008 |
| Barack | 2008 |
| Barack Obama | 2008 |
| D | 2008 |
| D | 2012 |
| St | 2008 |
| St | 2012 |
| Stat | 2008 |
| State | 2008 |

Figure 2: baseline

| Entity | Temporal exp. |
|---|---|
| Barack Obama | 2008 |
| Barack Obama | 2004 |
| Barack Obama | 2012 |
| States | 2008 |
| Obama | 2008 |
| Obama | 2004 |
| Obama | 2012 |
| Obama | 2010 |
| Obama | 2009 |
| United | 2008 |

Figure 3: our model

Figure 2 , 3 demonstrate the top 10 results from the baseline model and our model based on dataset 1. Our model clearly extracts more meaningful pairs than the baseline.

### 4.2 dataset 2

Our Baseline model performs very poorly on this dataset, in the retrieved top 50 pairs, none of them are correct. The main issue is that the baseline model does not prune the meaningless entities, which results in all the 50 pairs contain meaningless entities. Our proposed model retrieved 13 correct pairs out of the top 50, The improvement mainly comes from the syntactic feature (entities pruning).

| Entity | Temporal exp. |
|---|---|
| R | PRESENT_REF' |
| B | PRESENT_REF' |
| D | PRESENT_REF' |
| U | PRESENT_REF' |
| E | PRESENT_REF' |
| B | PAST_REF' |
| R | PAST_REF' |
| D | PAST_REF' |
| St | PRESENT_REF' |
| Re | PRESENT_REF' |

Figure 4: baseline

| Entity | Temporal exp. |
|---|---|
| United States | 1992 |
| Reagan | PT1S' |
| Senate | 2018-12-13 |
| United | 2018-12-13 |
| United | 1984 |
| United | 1992 |
| United | 1980 |
| United | 1996 |
| United | 1972 |
| United | PRESENT_REF |

Figure 5: our model

As shown by Figure 4 , 5. The top 10 results from baseline is dominated by meaningless entities. After applying the lexical and syntactic features, our model can extract some correct pairs in the top 10 results.

### 4.3 Error Analysis

The most frequently captured errors could be categorized into the poor performance of nltk name entity recognizer. A great deal of the entities extracted by nltk are actually confusing. One situation happened in our experiment is that the a single English character D is considered as an entity by nltk, such entities will have high probability of co-occurrence with other temporal expressions and will be extracted by our model incorrectly, we believe our model can perform better if a better name entity recognizer is given.

## 5 Conclusion

The current task focused on investigating the relations of entities and temporal expressions in text. More specifically, we explored how to extract the important pairs of entities and temporal expressions from plain text. Given the English plain text, we used the nltk (Bird et al., 2009) name entity recognizer to extract all the candidate entities, and we use SuTime (Chang and Manning, 2012) to extract all the candidate temporal expressions. All the pairs are ranked based on a linear combination of our lexical feature and syntactic feature. We constructed two dataset to evaluate our model, the result shows that our model performs well on the dataset with smaller size, when the dataset becomes larger and more complex, the performance of our model seems to be limited by the quality of entities extracted by nltk, but it is still significantly better than the baseline model.

## Acknowledgments

First and far most, we would like to thank prof. Davood Rafiei for the advice and help on this project, also special thank to Michael Su for helpful discussions.

## References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian*

*Semantic Web Conference*. Springer-Verlag, Berlin, Heidelberg, ISWC'07/ASWC'07, pages 722–735. http://dl.acm.org/citation.cfm?id=1785162.1785216.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, Beijing. https://doi.org/http://my.safaribooksonline.com/97805.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. ACM, New York, NY, USA, SIGMOD '08, pages 1247–1250. https://doi.org/10.1145/1376616.1376746.

Angel X. Chang and Christopher D. Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In *In LREC*.

Erdal Kuzey, Vinay Setty, Jannik Strötgen, and Gerhard Weikum. 2016. As time goes by: Comprehensive tagging of textual phrases with temporal scopes. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, WWW '16, pages 915–925. https://doi.org/10.1145/2872427.2883055.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, pages 55–60. https://doi.org/10.3115/v1/P14-5010.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '09, pages 1003–1011. http://dl.acm.org/citation.cfm?id=1690219.1690287.

Feng Niu, Ce Zhang, and Jude W Shavlik. 2012. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. volume 12(1).

Jannik Strötgen and Michael Gertz. 2012. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *LREC*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *16th International Conference on the World Wide Web*. pages 697–706.