Using machine learning to predict temporal orientation of search engines' queries in the Temporalia challenge

Michele Filannino
School of Computer Science
The University of Manchester
M13 9PL, Manchester, UK
filannino.m@cs.manchester.ac.uk

Goran Nenadic
School of Computer Science
The University of Manchester
M13 9PL, Manchester, UK
g.nenadic@manchester.ac.uk

ABSTRACT

We present our approach to the NTCIR-11 Temporalia challenge, Temporal Query Intent Classification: predicting the temporal orientation (present, past, future, atemporal) of search engine user queries. We tackled the task as a machine learning classification problem. Due to the relatively small size of the training set provided, we used temporal-oriented attributes specifically designed to minimise the features' sparsity. The best submitted run achieved 66.33% of accuracy, by correctly predicting the temporal orientation of 199 test instances out of 300. We discuss the results of the manual error analysis performed on the predicted classes, which sheds light on the main sources of error. Finally, we present some a-posteriori improvements to the best submitted run, which lead to a 6% improvement in terms of accuracy (72.33%).

Team Name

UniMAN

Subtasks

Temporal Query Intent Classification (TQIC)

Keywords

Temporal IR, Search engine, Feature engineering, Machine learning, ManTIME

1. INTRODUCTION

Temporal information extraction [10, 11, 9] is pivotal for many Natural Language Processing (NLP) applications such as question answering, text summarization and machine translation. The use of such information also plays a crucial role in the field of Information Retrieval (IR).

Research in this context has lead to IR systems which consider temporal information of indexed documents and users' queries to improve their accuracy by temporally filtering results in order to better capture user's intent. Being able to predict the temporal orientation of a query like `'weather in manchester'`, makes search engines able to show updated real-time meteorological information, whereas in the case of `'Weather forecast manchester'` they are more likely to show results about the immediate future. Some queries (e.g. `'sunday times'`, `'galileo Galilei'`), on the other hand, do not have a specific temporal orientation.

To address this issue, a shared task (called Temporalia [7]) was organized by the Japanese National Institute of Informatics (NII) in which systems are asked to automatically predict the temporal orientation of a given user query in one of the following categories: past, present, future and atemporal.

Search queries are atemporal when they do not have a temporal intent. Therefore the corresponding search results are in principle not expected to change due to the passing of time. On the other hand, search results for past, recency and future queries are related to time. Recency queries refer to present events, future queries refer to predictions or scheduled events, and past queries are related to events already happened.

This paper describes how we tackled this problem. Section 2 introduces the characteristics of the data provided by the challenge organisers. Section 3 illustrates the proposed machine learning-based methodology along with the attributes explicitly designed to minimise features' sparsity. The Results section (4) presents the accuracy of the different submitted runs, investigates the main sources of error, and presents some further a-posteriori improvements to our best performing model. We conclude the paper with a Discussion section (5) and Conclusions (6).

2. DATA

The organisers of the Temporalia challenge released a data set of 80 search engine queries where each one consisted of the textual representation (query), the submission time and the gold temporal orientation class (`atemporal`, `past`, `recency` or `future`).

They also provided a set of 20 queries to be used as a preliminary test set, therefore without temporal orientation (unlabelled). We manually annotated them and once the organizers confirmed the quality of the annotation (95% of accuracy, 19/20 correctly classified) we included them in the training set.

The official benchmark test set for the challenge consisted of 300 unlabelled queries. The Table 1 shows an excerpt of the training data.

3. METHODOLOGY

The task can naturally be seen as a 4-class classification problem since each query is associated with one and only one class. We therefore tackled it using a supervised machine learning-based approach. We mostly focussed our work on designing and testing a set of temporal-related attributes with a small set of possible values. As a consequence, this allowed us to minimise the total number of features required to model the classification problem.

ID	Object	Attribute	$\|V\|$	Example: query/submission → attribute value
1	Q	Is it a Wikipedia page title?	2	"New York Times" → 'YES'
2	Q	Does it contain a temporal expression?	2	"june 2013 movies" → 'YES'
3	S	Submission's term	3	"Feb 28, 2013 GMT+0" → 'B'
4	S	Submission's trimester	4	"Aug 26, 2013 GMT+0" → 'M2'
5	B	Timing	4	"Movies 2012", "Feb 28, 2013 GMT+0" → 'past'
6	Q	Most frequent trigger class	5	"peso dollar exchange rate" → 'present'
7	Q	Wh type	5	"how did hitler die" → 'how'
8	Q	Most frequent TempoWordNet class	5	"current stock prices" → 'present'
9	Q	Most frequent POS tag tense	7	"what is stop kony 2012" → 'VBZ'
10	Q	Most frequent coarse-grained POS tag	8	"kony 2012 fake" → 'N'
11	Q	Trigger classes footprint	11	"what was I thinking lyrics" → 'past-atemporal'
12	B	Temporal Δ between submission and query	16	"father's day 2010", "Feb 28, 2013 GMT+0" → 36.0
13	Q	Tenses footprint	18	"when does fall start" → 'VBZ-VB'
14	Q	Ordered TempoWordNet classes	18	"the last song" → 'past-future-present-atemporal'
15	Q	Most frequent fine-grained POS tag	21	"kony 2012 fake" → 'NN'
16	Q	Coarse-grained POS tag ordered footprint	119	"when is labour day" → 'N-W-V'
17	Q	Fine-grained POS tag ordered footprint	202	"when is labour day" → 'NN-WRB-VBZ'
18	Q	Coarse-grained POS tag footprint	204	"when is labour day" → 'W-V-N-N'
19	Q	Fine-grained POS tag footprint	265	"when is labour day" → 'WRB-VBZ-NN-NN'

Table 2: List of attributes used ordered by number of possible values. Object column indicates whether the attribute is computed from the (Q)uery, the (S)ubmission date or (B)oth. The $|V|$ column contains the cardinality of the value set per attribute (measured on the entire data set). Coarse-grained POS tags have been computed by considering just the first letter of the *Penn Treebank Tag Set*. POS tags are computed using the MaxEnt Treebank POS tagger from the Python NLTK library. The TempoWordNet-related attributes (#8 and #14) use the WordNet-based lemmatiser.

Query	Submission	Class
Movies 2012	Feb 28, 2013	Past
Upcoming Movies in 2013	Jan 1, 2013	future
2013 MLB Playoff Schedule	Jan 1, 2013	future
current price of gold	Feb 28, 2013	recency
Amazon Deal of the Day	Feb 28, 2013	recency
Number of Neck Muscles	Feb 28, 2013	atemporal
...	...	...

Table 1: Example of the training instances. All the queries have been submitted at GMT+0.

While *attribute*, *feature* and *value* are often used synonymously, in this paper we use them with a definition mutuated from the machine learning community [1]. In particular, a feature F is a true predicate expressing the pairing of a particular attribute h and its value v. For example, `lower=upcoming` is a feature, where `lower` is the attribute and `upcoming` is its value.

3.1 Pre-processing

All the user queries from the training and test data have been firstly pre-processed: for each user query we computed its lower-case version, its tokenisation and POS tags. The submission times have been pre-processed too: for each of them we firstly normalised[1] it via NorMA [4] (a temporal expression normaliser), and from this we separately extracted the numerical representation of year, month and day. The time of the query submission has not been taken into account.

[1]A temporal normaliser provides a standard ISO-8601 representation of any temporal expression: dates, durations, times and sets.

3.2 Attributes

The limited size of the training set made the task challenging for machine learning since the use of the attributes commonly used in NLP would have easily lead to sparse feature space, potentially leading to high-variance models (overfitting) in a real search engine's use scenario. By using just 100 samples, bag-of-words and n-grams representations would not have provided any support due to the huge number of possible different values to be learned.

We proposed 19 different attributes each one with a different number of possible values. An overview of them, along with explanatory examples is presented in Table 2.

Sometimes search engines are used as a faster alternative to typing the precise URL of our preferred web sites. This is the case, for example, of queries such as "the sunday times" or "wikipedia". We introduced the attribute #1 (see Table 2) to capture such cases. The titles of all the Wikipedia English pages have been collected via DBPedia [2]. The attribute value is positive only if a Wikipedia title and the query (as it is) are case-insensitively equal.

The information about the presence of temporal expressions in the query text (attribute #2) is important to separate the atemporal queries from the rest of them. We used ManTIME [5], a temporal expression extraction system, to extract the temporal expressions from the queries' text. We also used a backup regular expression-based system to spot date mentions (e.g. *"2012"*, *"1900"*), only in the case ManTIME does not find any temporal expression. The attribute has a positive value only if at least one temporal expression, or date mention, has been extracted.

Via a preliminary analysis of the training data we noticed that the part of the year in which the query has been submitted could play a crucial role in the classification task. Consequently, we designed two attributes (#3 and #4). The first one assigns 'B', 'M' or 'E' if respectively the query

ID	Attribute	Run 1	Run 2	Run 3
1	Is it a Wikipedia page title?		✓	✓
2	Does it contain a temporal expression?	✓	✓	✓
3	Submission's term			✓
4	Submission's trimester			✓
5	Timing	✓	✓	✓
6	Most frequent trigger class	✓		✓
7	Wh type		✓	✓
8	Most frequent TempoWordNet class			✓
9	Most frequent POS tag tense	✓	✓	✓
10	Most frequent coarse-grained POS tag		✓	✓
11	Trigger classes footprint	✓	✓	✓
12	Temporal Δ between submission and query		✓	✓
13	Tenses footprint		✓	✓
14	Ordered TempoWordNet classes			✓
15	Most frequent fine-grained POS tag		✓	✓
16	Coarse-grained POS tag ordered footprint			✓
17	Fine-grained POS tag ordered footprint			✓
18	Coarse-grained POS tag footprint			✓
19	Fine-grained POS tag footprint			✓

Table 3: List of attributes used in the submitted runs with reference to Table 2.

has been submitted in the first, second or third term (four-month period) of the year. The second one uses trimesters instead, leading to 4 possible values: 'B', 'M1', 'M2' or 'E'. Table 2 provides some examples. Using the normalised submission time and the extracted temporal expressions from the query text, we also compute two supplementary temporal attributes: #5 and #12. The latter is a numerical attribute corresponding to the difference, in terms of months, between the temporal expressions in the query and the submission date. The attribute #5 represents just its "sign" in the following categories: `present`, `past`, `future`.

From the training data we extracted the word and bigram vocabulary of the queries and filtered them as attributes by using RELIEF [8], a feature selection algorithm. We have been able to obtain a ranked list of the most (and least) influential unigrams and bigrams with respect to the classification task. Through a manual analysis, we grouped them in temporal trigger gazetteers, one per temporal class, according to their pertinence. For example the future triggers include words such as *"forecast"*, *"upcoming"*, whereas the past triggers include words such as *"last"* and *"previous"*. The attribute #6 represents the most frequent temporal trigger type in the query, whereas the attribute #11 represents the entire sequence of triggers in the order they appear in the query ("footprint").

We integrated TempoWordNet [3], a lexical knowledge-base for temporal analysis which provides a probabilistic measure of temporal orientations for the WordNet's synsets. Since WordNet's synsets are sets of lemmas we lemmatised the search query and represented the most likely temporal orientation class according to TempoWordNet (attribute #8) and the sorted list of them (attribute #14). For each lemma, the most likely corresponding WordNet sense has been used.

We also checked if a query is a wh-question. The attribute #7 represents which type of question the query is among the following possibilities: "what", "when", "where", "who", "why", and "how". The attribute just checks the query's first word.

Since queries are usually small multi-word expressions, we investigated the use of POS tags in different ways. The rationale was that specific sequences of tags could be correlated with some classes. The attributes #9 and #13, in particular, are focussed on verbs only. They represent the most frequent POS tag tense and the entire footprint, respectively. Attributes #10 and #15 are the most frequent coarse and fine-grained POS tag, respectively. Finally, the last four attributes (#16-19) are POS tag footprints ordered by the frequency or by order of appearance, using coarse and fine-grained tags.

For each of the attributes presented we also counted the cardinality of their value sets ($|V|$ column in Table 2): the number of different values each attribute can take. The counts have been computed using the entire data set (training and test) and it provides a rough, but useful, estimation of their sparsity.

3.3 Submitted Runs

We experimented with different machine learning models: SVM with linear, polynomial and RBF kernel, Naïve Bayes, C4.5 decision tree and Random Forests. The parameters for SVMs have been preliminary optimised on a sub set of the training data (20 samples) and 10 cross-fold validation has been used for all the experiments. We noticed the SVM (with polynomial kernel) and Random Forest algorithm systematically outperforming the rest. We used the former in Run 1 and 2, and the Random Forest algorithm for the Run 3. The attributes used for each run are illustrated in Table 3.

For the Run 1, called `minimal`, we selected the first 11 attributes and discarded the ones that did not positively contribute to the model (measured with RELIEF). In particular, we registered no improvements in the use of TempoWordNet-based attributes (#8 and #14), as well as the ones related to the submission part of the year (#3 and #4). The second run, called `intermediate`, is built on top of the first one, except for the absence of the most frequent trigger classes (#6). We added all the features with a cardinality less than 100, except for the TempoWordNet-related ones. The third run, called `full`, uses all the attributes presented in Section 3.2.

4. RESULTS

Run 1 obtained the highest accuracy by correctly predicting the temporal orientation of 199 queries (66.33%) out of 300. The intermediate and full models achieved, as predicted, lower accuracy.

Run	Name	Accuracy	#
1	**minimal**	**66.33%**	199
2	intermediate	61.33%	184
3	full	55.00%	165

Table 4: Results of the three submitted runs. Attribute set names, accuracies and number of correctly predicted instances are shown.

In the challenge, the Run 1 ranked 5th among the best runs, and 11th out of the 17 submitted runs. Further analysis on the submitted models showed that there is no statistically significant difference between the minimal and intermediate model. On the contrary, there is a statistically significant difference between minimal and full, and intermediate and full.

4.1 Error analysis

An analysis of the confusion matrix for the minimal run (see Table 5, below) highlights interesting issues.

	Classified as:			
	Recent	*Past*	*Future*	*Atemporal*
Recent	**43**	0	*21*	11
Past	3	**60**	6	6
Future	*38*	0	**35**	2
Atemporal	6	5	3	**61**

Table 5: Confusion matrix of the minimal run predictions for the official benchmark test set. True positive diagonal is in bold. Problematic cases are italicized.

We are able to identify three different major sources of classification mistakes, presented by their frequency:

Future as recent 38 `future` instances have been misclassified as `recent`. Some example of misclassified queries are: *"college rankings in 2013"*, *"2013 wimbledon"* and *"voice 2013 winner"*, which have all been submitted on the 1st of May 2013. The events described in the queries did not happen yet at the time of search and therefore the temporal orientation should have been future.

Recent as future. In 21 cases, `recent` instances have been misclassified as `future`. Some examples of misclassified queries are: *"bruins game tonight time"*, *"weather for nyc"*, *"when does spring start 2014"* submitted on 1st of May 2013. The first two examples are clear cases of recent temporal orientation, since the user is searching for information related to the day of the search. The last example, on the contrary, is questionable: the query could have been annotated as atemporal since the information searched for does not dependent on time.

Recent as atemporal. Finally, 11 `recent` instances have been erroneously classified as `atemporal`. Some examples of misclassified queries are: *"value of silver dollars"*, *"time in hawaii"*, *"24 hour clock"*, and *"disney prices going up"*. In all these cases the users expect search results which are strictly related with the current time. Prices, currency values and updated times are all examples of such category.

By manually investigating the attribute representation of these errors, we found that the major part of them are due to the absence of some trigger words in the gazetteers. In some cases, the misclassification is due to a wrong grouping of triggers. For example, the trigger *"tonight"* has been assigned to the future gazetteer instead of the recency one. Only a small part of them is due to the classifier limitations.

More generally, we also find some limitations in the representation of attributes, which if solved could have lead to better classification performance. Multi-valued attributes (#11 and #14) could have been substituted with groups of binary features. Some attributes (#10, #13 and #15-17) were affected by some ordering problems, which lead to different string representation though conveying the same information. Due to the choice of attributes selected for the best run (minimal) only the wrong trigger classification (#11) affects the best performance.

4.2 A-posteriori improvements

By fixing the limitations mentioned above, the minimal model correctly classifies 217 instances (18 instances more) of the official benchmark test set, achieving an accuracy of 72.33% (+6%).

By using the corrected attribute set only, we also determined which model would have provided the best performance. An exhaustive search among all the possible combinations of attribute sub-sets found the best of them providing 76% of accuracy (228 instances correctly classified). This level represents the upper bound for the accuracy of our attribute sets on the official benchmark test instances.

5. DISCUSSION

We found that the task proved to be challenging due to some specific characteristics. The most important one is the dimension of the training set. We believe that 100 instances are surely not enough to train a robust machine learning classifier, due to the fact that many of the classic NLP attributes in the literature have a too sparse representation to be learned from such a small training set. At the same time, we perceive this limitation as a deliberately conceived characteristic of the data intended to avoid overfitting attributes/rules, which would have ultimately resulted in no future use for the community.

During our manual error analysis, we also found that some of the queries were particularly hard to classify even for people. An example is *"Ventura Stern 2016"* which refers to the nominee of a comedian duo to the 2016 USA elections. Some other queries were just partial (e.g. *""earth after 1"*). In some other cases, we faced the need of surfing the Internet to seek some temporal information about entities mentioned in the queries. This has been the case for *"season 2 dexter"* or *"season 3 game of thrones"*, which both refer to particular seasons of famous TV shows. These findings suggest a potential benefit from the use of a named-entity recogniser

component along with some temporal contextualisation of the recognised concepts [6].

Finally, we found the contribution of TempoWordNet (as used by our attributes) to be negligible. The reason is that the temporal orientation of a word is related to its WordNet sense rather than its word-form which was essential in our task. Temporal orientation of all the verbs, for example, are inevitably missed since verbs in WordNet are represented through their infinitive form only. This also leads to a distribution of temporal orientation among senses which is skewed towards the atemporal class. 81.97% of senses have high probability of being atemporal, 13.72% of being present, 2.84% of being future, and just 1.48% of being past. If the atemporal label, and to some extent the present label too, can be seen as a neutral choices, lots of examples from future and past categories seem not to have any relation at all with the temporal orientation of the sense.

6. CONCLUSIONS

In this paper we presented our approach to the Temporal Query Intent Classification subtask of Temporalia in the NTCIR-11 challenge. We tackled the task as a machine learning classification problem, by designing and proposing a set of temporal-oriented attributes which minimised the features' sparsity. An extensive overview of the attributes used, along with examples, has been illustrated in Section 3.2.

Three different runs have been submitted, corresponding to three different attribute sets (minimal, intermediate and full) and two different machine learning classification algorithms (SVM with polynomial kernel and Random Forest). The minimal attribute set, which minimised the sparsity of the representation, achieved the best performance (66.33%) among our submitted runs. The model has been further improved, leading to a final accuracy of 72.33%.

A manual error analysis has been performed in order to highlight the main sources of classification error. We found that the major part of them are due to limitations related with the attribute representation.

To aid replicability of this work, the source code, the machine learning pre-trained models and the feature tables are available at `http://www.cs.man.ac.uk/~filannim/temporalia.html`. All the data are available for the submitted runs and the fixed one.

7. ACKNOWLEDGEMENTS

MF would like to acknowledge the support of the UK Engineering and Physical Science Research Council in the form of doctoral training grant. We want to thank Gaël Dias and Mohammed Hasanuzzaman for the availability in sharing with us TempoWordNet before its official public release.

8. REFERENCES

[1] S. Abney. *Semisupervised Learning for Computational Linguistics*, chapter 2, pages 14–15. Chapman & Hall/CRC, 1st edition, 2007.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[3] G. H. Dias, M. Hasanuzzaman, S. Ferrari, and Y. Mathet. TempoWordNet for sentence time tagging. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, WWW Companion '14, pages 833–838, Republic and Canton of Geneva, Switzerland, 2014. International World Wide Web Conferences Steering Committee.

[4] M. Filannino. Temporal expression normalisation in natural language texts. *CoRR*, abs/1206.2010, 2012.

[5] M. Filannino, G. Brown, and G. Nenadic. ManTIME: Temporal expression identification and normalization in the TempEval-3 challenge. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 53–57, Atlanta, Georgia, USA, June 2013. ACL.

[6] M. Filannino and G. Nenadic. Mining temporal footprints from Wikipedia. In *Proceedings of the First AHA!-Workshop on Information Discovery in Text*, pages 7–13, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.

[7] H. Joho, A. Jatowt, R. Blanco, H. Naka, and S. Yamamoto. Overview of NTCIR-11 Temporal Information Access (Temporalia) Task. In *Proceedings of the NTCIR-11 Conference*, 2014.

[8] K. Kira and L. A. Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Workshop on Machine Learning*, ML92, pages 249–256, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[9] N. UzZaman, H. Llorens, L. Derczynski, J. Allen, M. Verhagen, and J. Pustejovsky. SemEval-2013 Task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA, June 2013. ACL.

[10] M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky. SemEval-2007 Task 15: TempEval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 75–80, Prague, 2007.

[11] M. Verhagen, R. Saurí, T. Caselli, and J. Pustejovsky. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 57–62, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.