# KGO at the NTCIR-12 Temporalia Task: Exploring Temporal Information in Search Queries

Xin Kang[†]
kang-xin@is.tokushima-u.ac.jp

Yunong Wu[‡]
wuyunong@brandbigdata.com

Fuji Ren[†]
ren@is.tokushima-u.ac.jp

[†]Faculty of Engineering, Tokushima University
[‡]Department of Research and Development, Business Big Data Co., Ltd.

## ABSTRACT

This paper details our participation in the Temporal Intent Disambiguation (TID) English subtask of the NTCIR-12 Temporalia Task. Our work focuses on the development of a series of temporal features in web search queries and the construction of a deep neural network for disambiguating people's temporal intents in web searches. We analyze the importance of different temporal features and discuss the impact of neural network structures to the TID results.

## Team Name

KGO

## Subtasks

Temporal Intent Disambiguation (TID) Subtask (English)

## Keywords

Temporal Information Resolution, Temporal Feature Extraction, Deep Neural Network

## 1. INTRODUCTION

In the TID English subtask of the NTCIR-12 Temporalia Task [6], the KGO team carries forward their previous work in the Temporal Query Intent Classification subtask, as the previous TUTA1 team in the NTCIR-11 Temporalia Task. Our work focuses on the development of temporal features, in the sense of resolving temporal information in search queries, and the construction of a deep neural network, for disambiguating people's temporal intents in the web search with a probabilistic interpretation.

Although the explicit temporal features such as time gap and verb tense have been proved to be effective in classifying temporal query intents, they can be only observed in a very small part of web search queries in the real world [15]. For feature development, we take Wikipedia as a general knowledge base, for acquiring the temporal information about named entities in a search query. All the explicitly time-related information is extracted from the Wikipedia descriptions. We evaluate the correlation between a search query and the content of Wikipedia descriptions, to avoid taking the misleading information in TID.

We construct a deep neural network to solve the TID problem, based on our extracted temporal features. To generate probabilistic predictions of the queries' temporal intents, we employ the softmax activation function for neural network output. The neural network contains a dropout layer before each activation layer, which forces the model to learn robust feature abstractions during training, by randomly ignoring a fraction of the input units. The rest configurations of our deep neural network, including the number of hidden layers, the number of neurons in the input and hidden layers, the activation function for each hidden layer, and the batch size for fitting the model, are selected with a 5-fold cross-validation on the Dry Run data.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work in retrieving temporal intents in search queries. Section 3 describes our temporal feature extraction method in detail, and section 4 illustrates the construction of a deep neural network for TID. We report our experiment and analyze results in section 5, and conclude this paper in section 6.

## 2. RELATED WORK

The Temporal Query Intent Classification (TQIC) subtask in NTCIR-11 Temporalia task [9], as the previous challenge of TID, attracted 17 runs from 6 teams. The problem was to generate a temporal label for each query from Past, Recent, Future, and Atemporal. Most teams focused on the development of effective features for temporal intent classification. It turned out that raw words (or lemmas), verb tense (or part-of-speech tags) [7, 15, 1, 3], temporal dictionary [7, 5, 1, 3], and web search results [7, 5, 1, 3] were the most widely employed features, time gap [7, 15, 13] temporal expression [7, 13, 1], and named entities [7, 15, 1] were also actively explored, while features like query length [13] were less employed. Different classification algorithms were also employed for TQIC. However, because the training set only contained 100 samples [9] most classifiers suffered an over-fitting problem [15].

## 3. TEMPORAL FEATURE EXTRACTION

Web search queries are short. Fig. 1 plots the distributions of query length in the Dry Run and Formal Run sets of the TID and TQIC tasks. Because most search queries only contain a few words, the literal information in queries is usually not concrete enough for inferring people's temporal intents in web searches. For computers to understand people's temporal intents, the temporal information about searching events, which only exists in the human knowledge, must be explored.

Before diving into the feature extraction, we first look into some typical examples in the TID Dry Run set as shown below. The floating point numbers denote a probability dis-

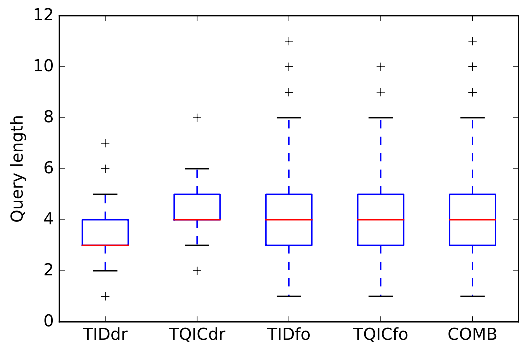


Figure 1: Distribution of query length. "dr" and "fo" are the abbreviations for Dry Run and Formal Run, respectively. COMB represents the combination of Dry Run and Formal Run data from TQIC and TID.

tribution of the temporal intent for a search query. We can easily tell the temporal label Past for a query like *beer night 1974* or *when was electricity invented*, either by comparing the explicit time description like the year *1974* with the query submission time *May 1, 2013*, or by inferring from the tense of English verbs like *was*. However, telling the temporal labels for *release date for iphone 6* or *memorial day* is not easy. We need to find the temporal information about *iphone 6* and *memorial day* elsewhere, and making predictions based on such information.

| Pa | Re | Fu | At | Time | Query |
|---|---|---|---|---|---|
| 1.0 | 0.0 | 0.0 | 0.0 | May 1, 2013 | beer night **1974** |
| 1.0 | 0.0 | 0.0 | 0.0 | May 1, 2013 | when **was** electricity invented |
| 0.0 | 0.0 | 1.0 | 0.0 | May 1, 2013 | release date for **iphone 6** |
| 0.0 | 0.0 | 0.6 | 0.4 | May 1, 2013 | **memorial day** |

We illustrate the development of a series of explicit and implicit temporal features in search queries in our participation of TID, which extends our previous work in TQIC by exploring Wikipedia as a knowledge base.

## 3.1 Explicit Temporal Features

The **Uppermost Verb Tense** feature indicates the tense of a query, which could be directly obtained by picking tense information from the main verb in a query. And since the main verb should always locate in a higher level than the other verbs in a syntactic tree, the tense information comes from the uppermost verb. As illustrated in our previous work [15], a verb tense can be implied through its part-of-speech label. For the example *when was electricity invented*, we pick part-of-speech tag of the uppermost most verb *was*, as shown in Fig. 2, to construct the temporal feature UVT_VBD. We employ the Stanford Parser library[14] in Stanford CoreNLP pipeline for syntactic parsing.

The **Time Gap** feature represents the difference between an explicit time expression like year *1974* and the query submission time. We extract the time information in queries with the SUTIME library[2] in Stanford CoreNLP pipeline. Time fields of year, month, day, season, and period in the time expressions are then normalized for differentiation with respect to the query submission time. We generate the temporal features of DIFF_past and DIFF_future for the

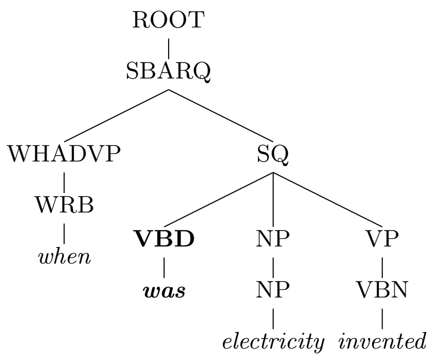


Figure 2: Syntactic tree for UVT feature extraction.

past and future time expressions, and generate features of DIFF_same_year, DIFF_same_month, DIFF_same_day, and DIFF_this for the same time expressions with different granularities. For example, DIFF_same_month is generated for time expression *May, 2013* with respect to the query submission time *May 1, 2013*.

## 3.2 Inexplicit Temporal Features

The **Temporal Named Entity** feature extracts temporal information of a named entity in the search query, by exploring Wikipedia as a knowledge base for the temporal information retrieval and by evaluating the semantic similarity between the search query and Wikipedia descriptions for the temporal correlation resolution.

We extract named entities in a query with the Stanford Named Entity Recognizer [4] in Stanford CoreNLP pipeline and with the TextRazor[1] Entity Extraction service. The knowledge about named entities are then obtained by querying Wikipedia for summaries with the Wikipedia API for Python[2]. For example, we get the Wikipedia summary of *iphone 6* as follows. *The iPhone 6 and iPhone 6 Plus are smartphones designed and marketed by Apple Inc. The devices are part of the iPhone series and were unveiled on September 9, 2014, and released on September 19, 2014. The iPhone 6 and iPhone 6 Plus jointly serve as successors to the iPhone 5C and iPhone 5S.* In the next step, we parse each sentence of the summary and extract Time Gap features. For *iphone 6*, we get DIFF_future from *September 9, 2014* and DIFF_future from *September 19, 2014*, respectively.

To resolve the correlation between a Time Gap feature and a named entity, we further extract their context information from the summary sentence and from the query respectively, and evaluate the similarity. Specifically, we extract the closest predicate to the Time Gap expression as its context based on the dependency parsing result from Stanford Parser, and extract the other words in the query except the named entity as its context. In the previous example, *unveiled* and *released* are extracted as the contexts of *September 9, 2014* and *September 19, 2014*, respectively, and *release date for* is extracted as the context of *iphone 6*. For each pair of the Time Gap feature $t$ and named entity $e$, we map their context words $w$ to $\mathrm{v}(w)$ in a 1000-dimensional semantic space based on a word2vec model [8, 10, 11, 12], which has been

[1]https://www.textrazor.com
[2]https://pypi.python.org/pypi/wikipedia

trained on the English Wikipedia (Feb 2015)[3], and calculate their context similarity as given below

$$s(t,e) = \cos\left(\frac{1}{|C_t|}\sum_{w\in C_t} \mathrm{v}(w), \frac{1}{|C_e|}\sum_{w\in C_e} \mathrm{v}(w)\right), \tag{1}$$

where $C_t$ and $C_e$ are the contexts for $t$ and $e$ respectively. In this case, we get the context similarities of 0.2511 and 0.3291 for resolving the correlations of two DIFF_future features with respect to the search query.

The **Holiday** feature explores the holidays as a special kind of named entity, and extracts their temporal information based on a holiday database[4]. Since some holidays are country-dependent, such as the *memorial day* in the United States, we extract the country information from queries with the TextRazor Entity Extraction and consider the United States as the default country on an naive assumption. And because some holiday dates are year-dependent, such as the *last Monday in May* for *memorial day* varies in each year, we further extract the year information from queries and consider the query submission year as a default. We query holiday date from the holiday database, with the holiday name, country, and year information, and differentiate it to the query submission time to get the DIFF_* features. In case of *memorial day* submitted on *May 1, 2013*, we get its holiday date of *May 27, 2013* and generate the DIFF_future feature.

### 3.3 Other Features

The **People** and **Time** features are the specific named entity categories for people and time, based on the TextRazor Entity Extraction results. For example, *Amy Winehouse* is recognized as deceased person, measured person, and person in the query *how did Amy Winehouse die*, while *Super Bowl* is recognized as recurring event in query *what time does the Super Bowl start*. These features are meaningful in the sense that they could combine different people or events together according to their semantic similarities, which effectively decrease the feature space to avoid over-fitting.

The **Lemma** feature normalizes words in a query, which is extracted with the Stanford Parser library. For each word in a query, we transform it to its dictionary form as a Lemma feature. Because the verb tense information would be lost in lemma, which is important for disambiguating people's temporal intents, we annotate the part-of-speech to verb lemmas. For example, *was* and *invented* are normalized as VBD_be and VBD_invent[5] in the query *when was electricity invented*.

## 4. NEURAL NETWORK CONSTRUCTION

We construct a deep neural network for solving the TID problem, with the temporal features extracted as described in section 3. The deep neural network is depicted in Fig. 3, with an input layer receiving the temporal features of a search query, an output layer generating the probabilistic predictions for four temporal categories, and several hidden layers combining the features into more abstract levels.

[3]https://github.com/idio/wiki2vec

[4]http://www.timeanddate.com/holidays/

[5]VBD is a wrong part-of-speech tag generated by the Stanford Parser. The correct one should be VBN.

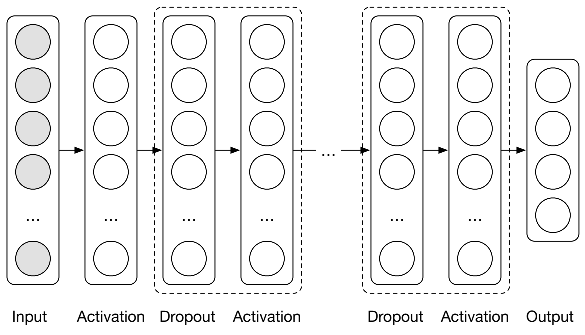


Figure 3: Deep neural network for TID.

A softmax activation is employed in the output layer for generating the probabilistic predictions $\tilde{p}$ as given below

$$\begin{aligned}\tilde{p}_j &= p(y=j|\boldsymbol{x};\boldsymbol{W}) \\ &= \frac{\exp(\boldsymbol{x}^T\boldsymbol{W}_j)}{\sum_{j'=1}^{J}\exp(\boldsymbol{x}^T\boldsymbol{W}_{j'})}.\end{aligned} \tag{2}$$

The softmax activation calculates the probability of assigning temporal category $j$ to the output $y$ given abstract features in $\boldsymbol{x}$ with a parametric weight matrix $\boldsymbol{W}$. $J = 4$ indicates the number of temporal categories for TID.

We employ cross entropy as the cost function

$$H(p,\tilde{p}) = -\sum_{j=1}^{J} p_j \log \tilde{p}_j. \tag{3}$$

The cross entropy $H(p,\tilde{p})$ is a non-negative value which describes the dissimilarity between the true distribution $p$ and predicted distribution $\tilde{p}$. $H(p,\tilde{p})$ gets its minimum value 0 if $\tilde{p}$ equals $p$. The training algorithm tries to minimize the mean of $H(p(i),\tilde{p}(i))$ for each training sample $i$, in order to make the neural network generate probabilistic predictions on temporal intents as close as to the training data.

We construct a deep neural network to solve the TID problem, in the sense that raw temporal features especially for the People, Time, and Lemma could be mapped into abstract features with a series of linear and non-linear filtering. This is achieved with multiple layers of hidden neurons in the deep neural network, as shown in Fig. 3. Each neuron of a hidden layer would receive the outputs of all neurons from its previous layer, and puts a weighted summation on these outputs before generating its own output based on an activation function. In the network, neurons on the same layer are similar in their function but receive the output of previous layer through different paths. Therefore, all neurons in a hidden layer put different weights for their weighted summation but keep the same activation function, which is selected from the activation set[6] of softplus, relu, tanh, sigmoid, hard sigmoid, and linear, based on a 5-fold cross-validation.

To avoid over-fitting the deep neural network, we add a dropout layer before each activation layer except the first one, which randomly ignores a fraction of outputs from the previous layer. During training, neurons in the activation layer activate on only a random fraction $q$ of the weighted summation which is filtered by the dropout layer, and try to generate the same predictions as before. During testing, the

[6]http://keras.io/activations/

dropout layer does not filter outputs from its previous layer but shrinks the values in $\boldsymbol{x}$ to $q\boldsymbol{x}$. The model is robust in the sense that it keeps a redundancy for feature abstraction in the hidden layers during training, which would save it from making wrong predictions for unseen samples during testing, on the assumption that redundant features in abstraction could provide crucial information for unseen samples for the final output.

## 5. EXPERIMENT

The TID subtask succeed the TQIC subtask in NTCIR-11, to predict the probabilities in 4 categories, i.e. Past, Recent, Future, and Atemporal, for analyzing people's temporal intents in web search queries. The difference between two subtasks is that TID allows multiple temporal labels in one query to reflect the possibilities of different interpretations while TQIC only allows one temporal label in one query. Because the data fields are transferable, we incorporate the TQIC data into the TID Dry Run set for training deep neural network, based on the assumption that there is no possibility mass for the temporal labels except the tagged one in TQIC data. And we use this model to generate the probabilistic predictions in the TID Formal Run set.

We employ the averaged per-class absolute loss score

$$\mathrm{loss}(p,\tilde{p}) = \frac{1}{J}\sum_{j=1}^{J}|p_j - \tilde{p}_j| \tag{4}$$

and the cosine similarity score

$$\mathrm{sim}(p,\tilde{p}) = \frac{\sum_{j=1}^{J} p_j \times \tilde{p}_j}{\sqrt{\left(\sum_{j=1}^{J} p_j \times p_j\right)\left(\sum_{j=1}^{J} \tilde{p}_j \times \tilde{p}_j\right)}} \tag{5}$$

for feature selection, model selection, and the final submission evaluation, where $p$ and $\tilde{p}$ are the true and predicted temporal probabilities of a query, and the subscript versions of $p_j$ and $\tilde{p}_j$ represent the true and predicted probabilities in temporal category $j$.

The feature selection is based on a simple neural network with only one hidden layer. We set 100 neurons in the hidden layer and the relu activation function for neurons in this layer. The model is fitted with a batch gradient descent algorithm, with 100 query samples in each batch. The TID results based on a 5-fold cross-validation in the training set indicates that feature combination of Uppermost Verb Tense, Time Gap, Temporal Named Entity, Holiday, People, and Lemma renders the best loss and cos scores. The Time feature is not included this feature combination, since the information in Time could have been covered by the Temporal Named Entity and Holiday features.

The parameter selection for deep neural network is based on sampling parameter values from the following uniform distributions

$$L \sim \mathrm{unif}(1,3), \tag{6}$$

$$\log_2(n^{(l)}) \sim \mathrm{unif}(2,7), \tag{7}$$

$$a^{(l)} \sim \mathrm{unif}(1,6), \tag{8}$$

$$\log_2(b) \sim \mathrm{unif}(6,9), \tag{9}$$

where $L$ denotes the number of hidden layers, $n^{(l)}$ is the number of neurons in layer $l$, $a^{(l)}$ is the index of activation function from {softplus, relu, tanh, sigmoid, hard sigmoid, linear}, and $b$ is the batch size in training. We also use early stopping to prevent over-fitting, by setting the number of epochs ($N$) in training based the averaged accuracy and loss scores, through a 5-fold cross validation.

We submit 3 Formal Run results based on the models selection performance.

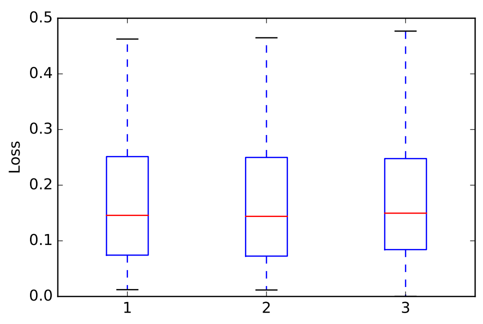
- KGO-TID-E-1: $L = 2$ hidden layers, with $n^{(1)} = 32$ and $n^{(2)} = 16$ neurons for each layer, activations of relu and hard sigmoid for each layer, batch size of $b = 256$, and $N = 385$ for the number of training epochs.
- KGO-TID-E-2: the same as KGO-TID-E-1, except $N = 392$ for the number of training epochs.
- KGO-TID-E-3: $L = 3$ hidden layers, with $n^1 = 64$, $n^2 = 32$, and $n^3 = 16$ neurons for each layer, activations of softplus, hard sigmoid, and linear for each layer, batch size of $b = 256$, and $N = 578$ for the number of training epochs.

The evaluation scores of our submissions are plotted in Fig. 4. Through 3 submissions, we find KGO-TID-E-2 with fewer layers rendering the best mean loss of 0.1676 and KGO-TID-E-3 with more layers rendering the best mean similarity of 0.8136. Besides, the maximum loss in KGO-TID-E-1 turns out lower than those in KGO-TID-E-2 and KGO-TID-E-3 and the minimum similarity in KGO-TID-E-1 turns out higher than those in KGO-TID-E-2 and KGO-TID-E-3. This suggests that the worst error of neural network in KGO-TID-E-1 is still better than the worst errors of the other neural networks. In addition, the minimum loss in KGO-TID-E-3 turns out lower than those in KGO-TID-E-1 and KGO-TID-E-2 and the maximum similarity in KGO-TID-E-3 turns out higher than those in KGO-TID-E-1 and KGO-TID-E-3. This implies that the best prediction of neural network in KGO-TID-E-3 is better than the best predictions of the other neural networks. KGO-TID-E-3 turns to have the largest range and the smallest interquartile range in both loss and similarity distributions, which suggests that it generates more predictions with the medium quality than KGO-TID-E1 and KGO-TID-E2, and a few predictions with the best and the worst qualities at the same time.

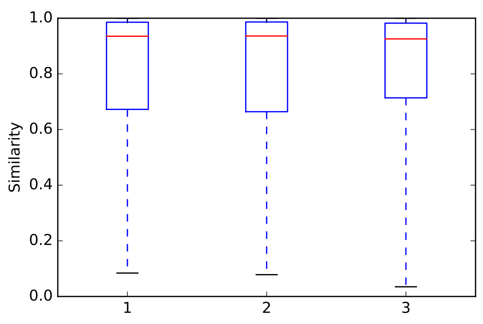
We plot the association between temporal features and temporal labels within the training corpus, as shown in Fig. 5. The association is evaluated by the normalized point-wise mutual information (npmi)

$$\mathrm{npmi}(x_i; y_j) = -\log\frac{p(x_i,y_j)}{p(x_i)p(y_j)}/\log p(x_i,y_j), \tag{10}$$

in which $x_i$ and $y_j$ are the temporal feature and temporal label, respectively. The subplots in the upper-left, upper-right, lower-left, and lower-right corresponds to Past, Recent, Future, and Atemporal labels. We integrate temporal features into categories, including the uppermost verb features (UVT), the verb tense feature (VT), the time gap features (TG), the named entity features (NE), and the lemma features (LM). For all temporal labels, many UVT features are centering around 0 npmi, since we use UVT_NULL for many queries which do not contain an uppermost verb. Keeping this feature might have confused the neural network. Most VT features diverge from 0 npmi, for Past, Future, and Atemporal, and the majority of TG and NE features diverge from 0 npmi for all temporal labels, suggesting the features are sensitive for these labels. The LM feature spreads in a



(a) Averaged per-class absolute loss.



(b) Cosine similarity.

Figure 4: Distribution of averaged per-class absolute loss and cosine similarity in 3 Formal Run submissions.

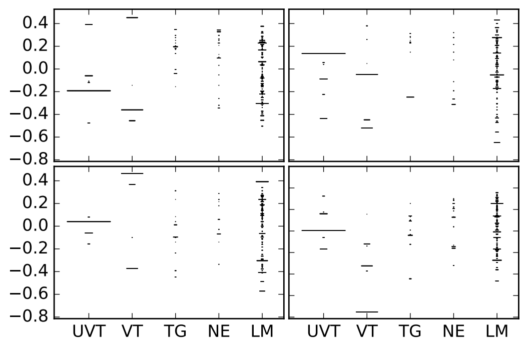


Figure 5: npmi between temporal features and labels.

wide npmi range, for all temporal labels, indicating that a finer investigation of lemmas with respect to the temporal labels could render further improvement.

## 6. CONCLUSION

In this paper, we described our participation in the Temporal Intent Disambiguation (TID) English subtask of the NTCIR-12 Temporalia Task. Because most queries contain very few words, the literal information in queries is usually not concrete enough for inferring people's temporal intents. Our work focused on exploring the temporal information underlying queries, based on the general knowledge from Wikipedia and the evaluation of context semantic correlations. The extraction of explicit temporal features, inexplicit temporal features, and other features has been thoroughly illustrated in this paper. We built deep neural network models to generate probabilistic distributions on 4 temporal intents, given these extracted temporal features in search queries. Dropout layers have been incorporated in the deep neural networks before each activation layer, which forces the model to learn robust feature abstractions in order to avoid over-fitting. The TID Dry Run data has been combined with the TQIC data in NTCIR-11 to construct a large training data set, based on which we selected the deep neural network structure through a 5-fold cross-validation and trained temporal intent prediction models. In our 3 submissions, KGO-TID-E-2 with fewer layers achieved the best loss score, and KGO-TID-E-3 with more layers rendered the highest similarity score. We also examined the association between different temporal features and temporal labels, and suggested some directions for improvement.

Our future work in TID would focus on learning of the temporal knowledge in natural language descriptions from Wikipedia and other sources of knowledge.

## 7. REFERENCES

[1] R. Burghartz and K. Berberich. Pi-inf at the ntcir-11 temporal query classification task. In *NTCIR*, 2014.

[2] A. X. Chang and C. D. Manning. Sutime: A library for recognizing and normalizing time expressions. In *LREC*, pages 3735–3740, 2012.

[3] M. Filannino and G. Nenadic. Using machine learning to predict temporal orientation of search engines' queries in the temporalia challenge. In *NTCIR*, 2014.

[4] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[5] M. Hasanuzzaman, G. Dias, and S. Ferrari. Hultech at the ntcir-11 temporalia task: Ensemble learning for temporal query intent classification. In *The 11th NTCIR Conference on Evaluation of Information Access Technologies*, pages p–478, 2014.

[6] J. Hideo, J. Adam, B. Roi, Y. Haitao, and Y. Shuhei. Overview of NTCIR-12 temporal information access (temporalia-2) task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, June 7-10, 2016, Tokyo, Japan*, 2016.

[7] Y. Hou, C. Tan, J. Xu, Y. Pan, Q. Chen, and X. Wang. Hitsz-icrc at ntcir-11 temporalia task. In *NTCIR*, 2014.

[8] Idio. Enwiki word2vec model 1000 dimensions. 2015.

[9] H. Joho, A. Jatowt, R. Blanco, H. Naka, and S. Yamamoto. Overview of ntcir-11 temporal information access (temporalia) task. In *NTCIR*. Citeseer, 2014.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[12] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751. ACL, 2013.

[13] A. Shah, D. Shah, and P. Majumder. Andd7@ ntcir-11 temporal information access task. In *NTCIR*, 2014.

[14] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. Parsing with compositional vector grammars. In *In Proceedings of the ACL conference*. Citeseer, 2013.

[15] H. Yu, X. Kang, and F. Ren. Tuta1 at the ntcir-11 temporalia task. 2014.