Understanding Temporal Query Intent

Mohammed Hasanuzzaman
Normandie University
UNICAEN, GREYC CNRS, France
mohammed.hasanuzzaman@unicaen.fr

Sriparna Saha
IIT-Patna
Kurji, Patna, India
sriparna@iitp.ac.in

Gaël Dias
Normandie University
UNICAEN, GREYC CNRS, France
gael.dias@unicaen.fr

Stéphane Ferrari
Normandie University
UNICAEN, GREYC CNRS, France
stephane.ferrari@unicaen.fr

ABSTRACT

Understanding the temporal orientation of web search queries is an important issue for the success of information access systems. In this paper, we propose a multi-objective ensemble learning solution that (1) allows to accurately classify queries along their temporal intent and (2) identifies a set of performing solutions thus offering a wide range of possible applications. Experiments show that correct representation of the problem can lead to great classification improvements when compared to recent state-of-the-art solutions and baseline ensemble techniques.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - Query Formulation

General Terms

Temporal IR, Ensemble learning, Multi-objective optimization

1. INTRODUCTION

As evidenced in [11], 1.5% of web search queries have explicit temporal intents (e.g. *fifa world cup 2014*) while the rate of implicit temporal queries (e.g. *history of coca-cola*) is more than 7%. Considering the large amount of queries issued everyday, recognizing the underlying temporal intent of a query is a crucial step towards improving the performance of search engines. For instance, this can be useful to select specific temporal retrieval models [10], temporally re-rank web results [9] or assess credible information [14].

Temporal query intent classification (TQIC), as first proposed in [8], seeks to determine whether users' information needs have a temporal dimension. This idea is pushed further in [6] and aims to determine whether the user is interested in information about the past, present or future when issuing a query or instead his information need has no temporal dimension.

Most recent works on TQIC have proposed supervised learning techniques based on the data set provided in NTCIR Temporalia [7]. Due to the small size of training data, best performing strategies have followed either the semi-supervised [16] or the ensemble learning paradigms [15, 5].

In this paper, we present an ensemble learning framework defined as a multi-objective optimization problem so to (1) obtain accurate classification results even when training evidences are limited and (2) identify different performing solutions thus offering a wide range of possible information access scenarios. Indeed, depending on the handled task, precise classification may be required (e.g. credible information assessment) or high recall may be preferred (e.g. temporal re-ranking).

Experiments over standard NTCIR data set show that great classification improvements (up to 16% accuracy) can be achieved compared to recent state-of-the-art solutions and baseline ensemble techniques, when a correct representation of the problem is proposed.

SIGIR'15 August 09 - 13, 2015, Santiago, Chile

DOI: http://dx.doi.org/10.1145/2766462.2767792.

2. RELATED WORK

The most influential attempt at temporal classification of queries comes from [8] who classify queries into three distinct classes: *atemporal*, *temporally unambiguous* and *temporally ambiguous*. In particular, they consider the distribution of retrieved documents over time and create meaningful features based on this distribution.

Other ideas for implicit temporal queries have been developed by [11]. By analyzing query logs, they investigate the automatic detection of implicitly year qualified queries, i.e. queries that refer to an event in a specific year without containing the year in the query string. Following the same motivation, [1] proposed a solution based on content temporal analysis. In particular, they identify top relevant dates in web snippets with respect to a given implicit temporal query and temporal disambiguation is performed through a distributional metric called GTE.

Recently, the NTCIR Temporalia task [7] pushed further this idea and propose to distinguish whether a given query is related to *past*, *recency*, *future* or *atemporal*. Within this context, the most performing system is based on a SVM semi-supervised learning algorithm [16] and uses the AOL 500K User Session Collection [12] as unlabeled data. Two other competitive systems [15, 5] rely on ensemble learning (especially majority voting). Indeed, due to the small size of training data (only 100 queries distributed equally by class), classification results are weak if a single classifier is used in a traditional supervised way. To overcome this situation, we follow the ensemble learning paradigm defined as a multi-objective optimization problem in a similar way as [13].

3. LEARNING INSTANCES FOR TQIC

TQIC can be defined as follows. Given a web search query q and its issuing date d, predict its temporal class $c \in \{past, recency, future, atemporal\}$. Based on this definition, [7] released a set of 100 training queries (25 queries for each class) and 300 testing queries (75 queries for each class). We will call this data set NTCIR-TQIC. Examples of the form $< q, d, c >$ are given as follows.

q	d	c
who was martin luther	Jan 1, 2013 GMT+0	*past*
amazon deal of the day	Feb 28, 2013 GMT+0	*recency*
release date for ios7	Jan 1, 2013 GMT+0	*future*
number of neck muscles	Feb 28, 2013 GMT+0	*atemporal*

3.1 External Resources

The NTCIR-TQIC data set evidences two crucial limitations. First, the training set is small. Second, the amount of literal features is limited as queries are short (between 3 and 4 words). The first case is a classical learning problem and it is discussed in section 4. As for the second case, external resources were used to expand the amount of query information [1].

So, for each query, we first collected the top K web snippets[1] returned by the Bing search API[2]. The underlying idea is that web snippets are likely to evidence temporal information if the query has a temporal dimension.

Then, for each query, we collected its most relevant year date along with its confidence value from the freely available web service GTE[3] proposed in [1]. In particular, given a temporally implicit query, GTE extracts from web snippets query-relevant year dates based on distributional similarity. Some examples of extracted year dates and confidence values are given as follows[4].

q	Most confident Year	Confidence value
who was martin luther	1929	0.944
amazon deal of the day	2015	0.760
release date for ios7	2013	0.893
number of neck muscles	2014	0.708

3.2 Features Definition

Most related works have proposed similar sets of features [7]. The most common are time gap, verb tense, named entities, lemmas and specific temporal words. Here, we identified 11 independent features from the query string, its issuing date and the extra data collected. In particular, we relied on previous studies and available tools and resources developed in our laboratory [1, 3]. Details of the different features are given as follows.

D_b_Dates: This feature aims to evaluate the time gap between the query and its issuing date. It is calculated as the difference between the year date explicitly mentioned in the query string q and the issue year date d_{year}. If there is no mention of a date inside q (timely implicit query), we consider the most confident year date obtained from GTE [1]. If no date is returned by GTE, this feature is given a null value.

C_o_Date: This feature aims to evidence the confidence value over the time gap definition. It is set to 1 when there is explicit mention of a year date inside q string (maximum confidence). Otherwise, it is set to the returned confidence value of GTE [1]. A 0 value is given if no date is returned by GTE.

[1]For computational reasons, we set $K = 10$.

[2]`https://datamarket.azure.com/dataset/bing/search`

[3]`http://wia.info.unicaen.fr/GTEAspNetFlatTempCluster_Server/`

[4]Extraction was processed April, 16th 2015 for illustration.

N_o_PW, **N_o_RW** and **N_o_FW:** These features aim to capture the query timeliness based on its temporal content words. They respectively represent the number of words in q belonging to past, present and future categories in TempoWordNet[5] [3].

N_o_PS, **N_o_RS**, **N_o_FS** and **N_o_AS:** This set of features aims to interpret the timeliness of the query q based on the temporality of its returned web snippets. The rationale is that if a query has a temporal dimension, web search results should evidence the same intent. So, for any q, these features are respectively the number of returned web snippets classified as past, recency, future and atemporal by the sentence temporal classifier (STC) defined in [3].

C_o_Q: The aim of this feature is to define the intrinsic temporality of a query (as if it was a sentence). So, this feature takes the value returned by STC [3] (i.e. past, recency, future or atemporal) when taking the query string q as input.

Q_S: The rationale of this feature is that specific (non-temporal) words may play an important role in temporal classification. As a consequence, each query string q is represented as its bag of unigrams where the presence of a word is associated to the value 1 and 0 when it is not present.

In order to better assess the importance of each individual feature, we present the top 5 features in terms of Information Gain (IG) in the following Table.

Feature	IG	Main resource/tool
D_b_Dates	0.245	GTE [1]
N_o_FW	0.219	TempoWordNet [3]
N_o_FS	0.135	STC [3]
C_o_Q	0.130	STC [3]
C_o_Date	0.114	GTE [1]

Results show that both the extra collected data (i.e. Bing web snippets and GTE year dates) and the different used resources and tools (i.e. TempoWordNet and STC) play an important role for the temporal query intent classification task.

4. LEARNING FRAMEWORK

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or binary voting) to classify new examples [4]. In particular, ensemble learning is known to obtain highly accurate classifiers by combining less accurate ones thus allowing to overcome the training data size problem. Many methods for constructing ensembles have been developed in the literature [4]. In this paper, we propose to define ensemble learning as a multi-objective optimization (MOO) problem. Our motivations are two-fold. First, [13] showed that MOO strategies evidence improved results when compared to single objective solutions and state-of-the-art baselines. Second, MOO techniques propose a set of performing solutions rather than a single one. As TQIC can be thought as an intermediate module in some larger application (e.g. retrieval, ranking or visualization), offering different performing solutions can be a great asset to adapt to any kind of information access situation without loss of reliability.

4.1 MOO Problem Defintion

A definition of multi-objective optimization can be stated as follows: find the vector $\bar{x} = [x_1, x_2, \ldots, x_n]^T$ of decision variables that optimize O objective functions $\{O_1(\bar{x}), O_2(\bar{x}), \ldots, O_O(\bar{x})\}$ simultaneously which also satisfy user-defined constraints, if any. The concept of domination is also an important aspect of MOO. In case of maximization, a solution $\overline{x_i}$ is said to dominate $\overline{x_j}$ if both conditions (1) and (2) are satisfied.

[5]In particular, we used TWnL available at `https://tempowordnet.greyc.fr/`.

$$\forall k \in 1,2,\ldots,O, \quad O_k(\overline{x_i}) \geq O_k(\overline{x_j}) \tag{1}$$

$$\exists k \in 1,2,\ldots,O, \quad O_k(\overline{x_i}) > O_k(\overline{x_j}) \tag{2}$$

Finally, the set of non-dominated solutions of the whole search space S is called the Pareto optimal front, from which a single solution may be selected based on any suitable criterion.

Ensemble learning can be seen as a vote based problem. Suppose that one has a total number of N classifiers $\{C_1, C_2, ..., C_N\}$ trained for a M class problem. Then, the vote based classifier ensemble problem can be defined as finding the combination of votes V per classifier C_i, which will optimize a quality function $F(V)$. V can either represent a binary matrix (binary vote based ensemble) or a matrix containing real values (real/weighted vote based ensemble) of size $N \times M$. In case of binary voting, $V(i,j)$ represents whether C_i is permitted to vote for class M_j. $V(i,j) = 1$ is interpreted as the i^{th} classifier is permitted to vote for the j^{th} class else $V(i,j) = 0$ is interpreted as the i^{th} classifier is not permitted to vote for the j^{th} class. In case of real voting, $V(i,j) \in [0,1]$ quantifies the weight of vote of C_i for the class M_j. If a particular classifier is confident in determining a particular class, then more weight should be assigned for that particular pair, otherwise less weight should be attributed.

In terms of MOO formulation, the classifier ensemble problem at hand is defined as determining the appropriate combination of votes V per classifier such that objectives $O_1(V)$ and $O_2(V)$ are simultaneously optimized and $O_1 =$ recall and $O_2 =$ precision.

4.2 Evolutionary Procedure

The multi-objective methods used here are based on the search capabilities of the non-dominated sorting genetic algorithm [2].

String Representation: In order to encode the classifier ensemble selection problem in terms of genetic algorithms, we propose to study three different representations.

(1) Simple Classifier Ensemble (SCE): each individual classifier is allowed to vote or not. The chromosome is of length N and each position takes either 1 or 0 as value,

(2) Binary Vote based Classifier Ensemble (BVCE): each individual classifier is allowed to vote or not for a specific class M_j. The chromosome is of length $N \times M$ and each position takes either 1 or 0 as value,

(3) Real/weighted Vote based Classifier Ensemble (RVCE): all classifiers are allowed to vote for a specific class M_j with a different weight for each class. The chromosome is of length $N \times M$ and each position takes a real value.

Fitness: Each individual chromosome corresponds to a possible ensemble solution V, which must be evaluated in terms of fitness. Let the number of available classifiers be N and their respective individual F-measure values by class F_{ij}, $i = 1 \ldots N, j = 1 \ldots M$ (i.e. F_{ij} is the F-measure of C_i for class M_j). For a given query q, receiving class M_j is weighted as in Equation 3 where the output class assigned by C_i to q is given by $op(q, C_i)$. Note that in the case of SCE, $V(i,j)$ is redefined as $V(i,.)$ and F_{ij} as $F_{i.}$.

$$f(q, M_j) = \sum_{i=1:N \& op(q,C_i)=M_j} V(i,j) \times F_{ij}. \tag{3}$$

Finally, the class of the query q is given by $argmax_{M_j} f(q, M_j)$. As such, classifying all queries from a development set gives rise to two fitness (or objective) values, which are respectively recall (O_1) and precision (O_2) and must be optimized simultaneously.

Optimization and Selection: The multi-objective optimization problem is solved by using the Non-dominated Sorting Genetic Algorithm (NSGA-II) [2]. The most important component of NSGA-II is its elitism operation, where the non-dominated solutions present in the parent and child populations are moved to the next generation. The chromosomes present in the final population provide the set of different solutions to the ensemble problem and represent the Pareto optimal front.

It is important to note that all the solutions are important, representing a different way of ensembling the set of classifiers. But for the purpose of comparison with other methods, a single solution is required to be selected. For that purpose, we choose the solution that maximizes the F-measure based on its optimized sub-parts recall and precision as shown in equation 4.

$$F\text{-measure} = \frac{2 \times recall \times precision}{recall + precision}. \tag{4}$$

5. EXPERIMENTS

Experiments are run over a two-steps process. First, $N = 28$ individual classifiers are learned using 10-fold cross validation[6] over a subset of 80 training instances (20 examples for each of the $M = 4$ classes) randomly selected from the initial training set of NTCIR-TQIC containing 100 queries. For each classifier C_i, $F_{i.}$ (global F-measure) and F_{ij} (F-measure for class M_j) values are stored. All experiments were run over the Weka platform[7] with default parameters. Following Weka's denomination, the list of the 28 classifiers is as follows: NaiveBayes, NBTree, NNge, AdaBoostM1, Bagging, BayesNet, BFTree, ClassificationViaRegression, DecisionTable, FT, J48, JRip, IB1, IBk, Kstar, LWL, LMT, Logistic, LogitBoost, MultiBoostAB, MultilayerPerceptron, RandomCommittee, RandomForest, RBFNetwork, REPTree, RotationForest, SimpleLogistics and SMO. In order to assess the quality of each individual classifier, each one was tested on the NTCIR-TQIC test data set containing 300 unseen queries (75 for each class). Results of the top 5 classifiers are given in Table 1.

Classifiers	Precision	Recall	F-measure
Logistic	**82.9**	**75.0**	**78.8**
RandomForest	82.6	70.0	75.8
RotationForest	77.5	70.0	73.6
LMT	69.2	65.0	67.0
SimpleLogistics	69.2	65.0	67.0

Table 1: Results of single learning strategies.

The second step of the experiment is the optimization procedure. For that purpose, the remaining 20 query examples (5 for each class) from the NTCIR-TQIC training data set are used. We call it the development set. Based on the development set, the evolutionary optimization using NGSA-II is run for three representations (SCE, BVCE, RVCE) and the best solution is selected based on maximum F-measure as defined in equation 4. Performance results are presented in Table 2 and compared to two baselines ensemble techniques (BSL1, BSL2). BSL1 corresponds to Boosting with the single Logistic classifier and BSL2 is a SVM solution with 28 features each one corresponding to the output class (i.e. past, recency, future, atemporal) of each of the 28 classifiers.

As expected, our methodology outperforms BSL1 by 12.4% and BSL2 by 14.9% in terms of F-measure for the RVCE representation. In particular, BSL1 suffers from the use of a single classifier family while BSL2 can not generalize over the small amount of training data (only 20 examples). Moreover, the most fine tuned strategy in terms of ensemble learning evidences improved results

[6]Note that cross-validation is already an ensemble technique.

[7]`http://www.cs.waikato.ac.nz/ml/weka/`

Measures	RVCE	BVCE	SCE	BSL1	BSL2
Precision	**92.2**	85.1	86.0	82.9	77.5
Recall	**90.0**	85.0	83.7	75.0	75.0
F-measure	**91.1**	85.0	84.8	78.7	76.2

Table 2: Results of ensemble learning strategies.

when compared to coarse-grain solutions. Improvements of 6.1% and 6.3% are respectively shown against BVCE and SCE.

In order to understand the spectrum of the different solutions on the Pareto front, we present in Table 3 three different situations: the solution that maximizes precision (line 1), the solution that maximizes recall (line 2) and the solution that maximizes F-measure (line 3). Results show that high overall performances are provided by every solution. But, depending on the application at hand, one may expect to find a better tuned configuration.

Measures	Recall	Precision	F-measure
Max precision	88.7	**93.1**	90.9
Max recall	**90.8**	90.8	90.8
Max F-measure	90.0	92.2	**91.1**

Table 3: Precision and recall spectrum.

Finally, comparative accuracy results are given against state-of-the-art solutions from NTCIR-11 Temporalia TQIC in Table 4[8]. Our solution evidences highly improved results overall as well as for each individual class. In particular, accuracy improvements of 10% for past, 19% for recency, 9% for future, 10% for atemporal and 16% overall are achieved against best existing studies.

Classes	RVCE	#1 [16]	#2 [15]	#3 [5]
Past	**0.95**	0.85	0.75	0.79
Recency	**0.82**	0.48	0.56	0.63
Future	**0.94**	0.85	0.81	0.64
Atemporal	**0.89**	0.77	0.79	0.71
All	**0.90**	0.74	0.73	0.69

Table 4: Comparative accuracy results to state-of-the-art techniques presented in NTCIR-11 Temporalia task.

Note that all experimental results and data sets[9] of this paper are freely available at the following url `https://tempowordnet.greyc.fr` for reproducibility.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we tackled the problem of identifying temporal intent of queries from a machine learning point of view. Due to the small amount of gold training data, we proposed an ensemble learning solution, whose underlying idea is to reduce bias by combining multiple classifiers instead of relying on a single one. In particular, recently developed multi-objective based ensemble techniques have been applied to improve overall accuracy. For our purpose, we made use of a set of features which can easily be extracted from different freely available resources to allow reproducibility.
Initial results are interesting to us and open new avenues for future research such as to bring on the intent of temporal query into the cluster of semantically related web search results for better user satisfaction. We are also investigating how to assign multiple temporal classes to web search queries especially for the implicit ones, in the line of the next NTCIR-12 Temporalia task.

[8]Results are taken from [7].
[9]Where copyright issues do not apply.

7. REFERENCES

[1] R. Campos, G. Dias, A. Jorge, and C. Nunes. Gte: A distributional second-order co-occurrence approach to improve the identification of top relevant dates in web snippets. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2035–2039, 2012.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):181–197, 2002.

[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)*, pages 833–838, 2014.

[4] T. G. Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.

[5] Y. Hou, Q. Chen, J. Xu, Y. Pan, Q. Chen, and X. Wang. Hitsz-icrc at the ntcir-11 temporalia task. In *Proceedings of the NTCIR-11 Conference*, 2014.

[6] H. Joho, A. Jatowt, and R. Blanco. Ntcir temporalia: a test collection for temporal information access research. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion (WWW)*, pages 845–850, 2014.

[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] R. Jones and F. Diaz. Temporal profiles of queries. *ACM Transactions on Information Systems*, 25(3), 2007.

[9] N. Kanhabua and K. Nørvåg. Learning to rank search results for time-sensitive queries. In *Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM)*, pages 2463–2466, 2012.

[10] X. Li and W. B. Croft. Time-based language models. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM)*, pages 469–475, 2003.

[11] D. Metzler, R. Jones, F. Peng, and R. Zhang. Improving search relevance for implicitly temporal queries. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 700–701. ACM, 2009.

[12] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *Proceedings of the 1st International Conference on Scalable Information Systems (InfoScale)*, 2006.

[13] S. Saha and A. Ekbal. Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. *Data Knowledge Engineering*, 85:15–39, 2013.

[14] J. Schwarz and M. Morris. Augmenting web pages and search results to support credibility assessment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1245–1254, 2011.

[15] A. Shah, D. Shah, and P. Majumber. Andd7 @ ntcir-11 temporal information access task. In *Proceedings of the NTCIR-11 Conference*, 2014.

[16] H.-T. Yu, X. Kang, and F. Ren. Tuta1 at the ntcir-11 temporalia task. In *Proceedings of the NTCIR-11 Conference*, 2014.