GIR at the NTCIR-12 Temporalia Task

Long Chen, Joemon Jose, Haitao Yu, Fajie Yuan

School of Computing, University of Glasgow

Introduction: This slide describes our approach to solving the Temporal Intent Disambiguation (TID) problem and discusses the official results. We explore the rich temporal information in the labeled and unlabeled search queries. A semi-supervised linear classifiers is then built up to predict the temporal classes for each search query.

Methods: Our system consists of two separate modules: (a) identifying temporal features in search queries, and (b) exploring the contextual information in these queries. In addition to textual features, we find that there are also plenty of contextual information associated with each query available when submitting the query, such as verb tenses and temporal expressions returned by Google results, which provides different perspective to understand the query complementarily.

Classification of Query Intent

Textual Features: Textual pre-process is done by using Weka API. As for textual feature, we find that 3-gram characters and word have a better performance than other options. While 3-gram character can give a slightly better result, word is chosen as the textual representation as it is easier for people to have a better understanding of it.

Contextual Features: A total of 3 contextual features are learned in order to build the classifier. To capture the temporal features, we resort to the temporal expressions and verb tenses of a query. A temporal expression in a query is a sequence of terms that represent a point in time, a duration or a frequency. Verb tense is a specific indicator that signals a situation takes place.

Co-training: Given textual features and contextual features, a co-training algorithm is employed to exploit the power of unlabelled instances.

Datasets and Metrics: The Temporal Intent Disambiguation (TID) Subtask subtask provides a dry-run dataset of 100 search queries for developing the temporal classification models, and a formal-run dataset of 300 search query samples for examining the results. The details of the datasets and the metrics are described in the standard GIR overview paper.

Results

A number of machine learning algorithms implemented in Weka, including C4.5, Random Forest, Naive Bayes, k-Nearest-Neighbours, and Linear Support Vector Machine (SVM), have been tried out for semi-supervised learning (co-training). Linear SVM kept to deliver the best classification performance in our experiments, so we only report its results here. The system is evaluated against Averaged Per-Class Absolute Loss (APCAL) and Cosine Similarity. The formal run values our system achieved are 0.326 for APCAL and 0.417 for Cosine Similarity.

To test the effectiveness of our proposed model, the following three approaches are compared:

Supervised Approach, which simply employ the original dry run queries as training instances.

Semi-Supervised Approach, which is the method that we used in this slides, but without the enhancement of any external resources

Semi-Supervised Approach with external knowledge, which is similar to the second one but is equipped with the knowledge learned from Google

Table 1: Table 1: The experimental results on formal-run dataset.

measures	Supervised	Semi-Supervised	Semi-Supervised with External Knowledge
Cosine Similarity	0.366	0.384	0.417
Absolute Loss	0.416	0.358	0.326

Contact:

Lilybank Garden

School of Computing, University of Glasgow
United Kingdom

Ph: +4407514434466
Email: long.chen@glasgow.ac.uk