MPI-INF AT THE NTCIR-11 TEMPORAL QUERY CLASSIFICATION TASK

Robin Burghartz

Klaus Berberich

Max Planck Institute for Informatics,
Saarbrücken, Germany

General Approach

Overall strategy for TQIC subtask:

1. Focus on deriving features for classification
2. Rely on established off-the-shelf components to test a wide spectrum of different features

General Approach

Components:

- WEKA : for classification
- StanfordCoreNLP: for the various NLP processing needs of the task
- Hadoop/MapReduce: for the construction of a temporal dictionary

Overview

1. Feature Design
2. Results
3. Conclusion

Overview

1. Feature Design
2. Results
3. Conclusion

Feature Design - Overview

Different „classes“ of features:

- **Collection features** analyze temporal expressions and timestamps from the collection‘s documents
- **Linguistic features** consider linguistic properties of the query string
- **Trigger word features** allow the classifier to learn very clear time determiners (e.g. „forecast“, which hints to a future intent)
- **Semantic feature(s)** capture(s) the topic of a query

Collection Features

Collection Features

(1) Distribution of temporal expressions

What time do temporal expressions from pseudo-relevant documents refer to?

=> Content time

Collection Features

(1) Distribution of temporal expressions

Content time

Example:

d1: „2014“

d2: „January 2014“

Two principles:

- If temporal expression is imprecise, distribute weight
- Temporal expressions from more relevant documents get bigger weight in the distribution

Collection Features

(1) Distribution of temporal expressions

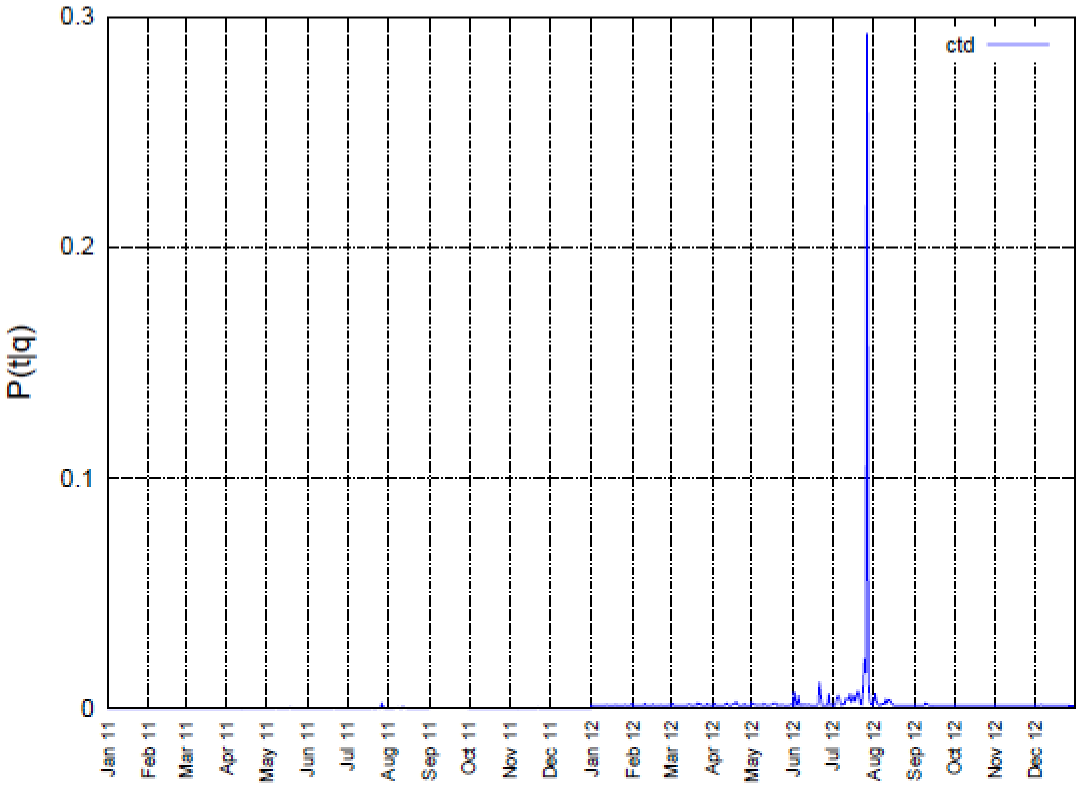

Figure 4.6.: CTD for Query "olympic games london"

Content time distribution, i.e. distribution of temporal expressions from relevant documents

Example Features:

- How much weight is before/on/after the query issue time?
- Where are the quantiles relative to the query issue time?

Collection Features

(1) Distribution of temporal expressions

Content time

$$P_{con}(t \mid q) = \frac{\sum_{d \in R} \sum_{te \in TE(d)} P(t \mid te) \cdot P(q \mid d)}{\sum_{d \in R} \sum_{te \in TE(d)} P(q \mid d)}$$

R: pseudo-relevant documents

$TE(d)$: set of temporal expressions from document d

Collection Features

(2) Distribution of timestamps

When were relevant documents published?

=> Publication time

$$\tilde{P}_{pub}(t \mid q) = \sum_{d \in R} P(t \mid d) \cdot \frac{P(q \mid d)}{\sum_{d' \in R} P(q \mid d')}$$

Collection Features

(2) Distribution of timestamps

Publication time:

$$\tilde{P}_{pub}(t \mid q) = \sum_{d \in R} P(t \mid d) \cdot \frac{P(q \mid d)}{\sum_{d' \in R} P(q \mid d')}$$

This approach is close to the approach used in „Temporal profiles of queries“ [Jones et Diaz] (similar classification task, but focus on the distinction between atemporal/temporal).

Collection Features

(2) Distribution of timestamps

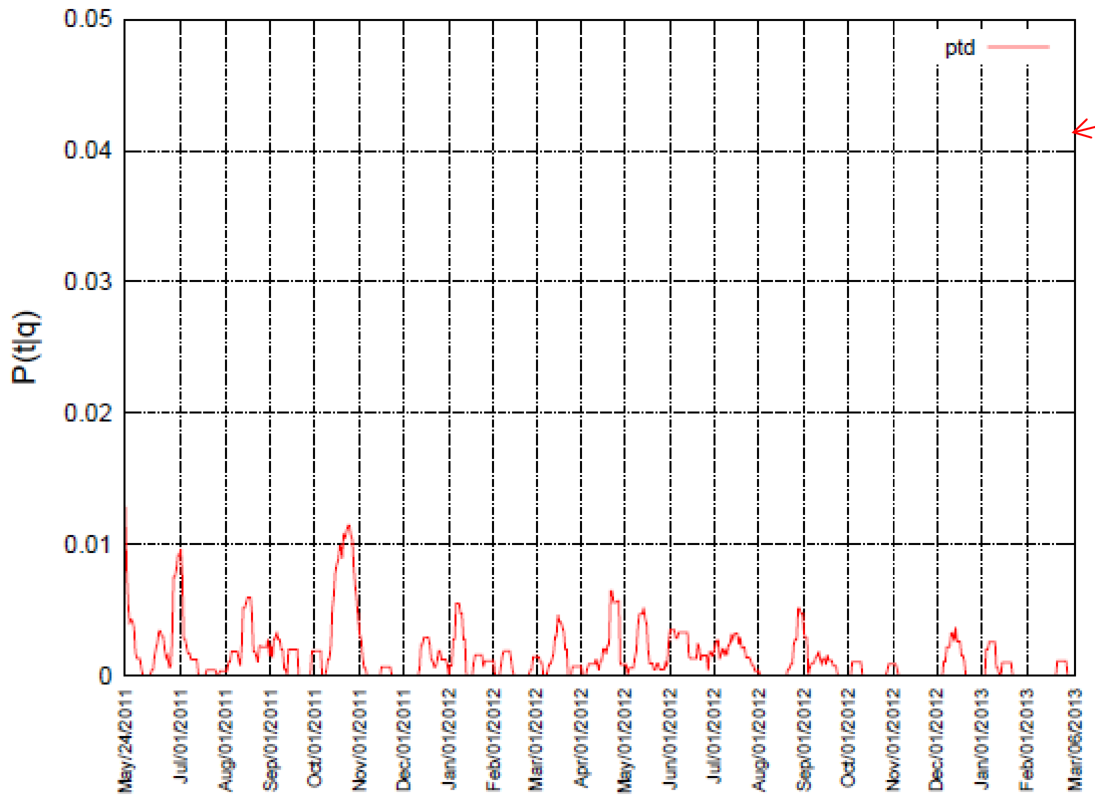

Figure 4.2.: PTD for Query "weather forecast"

Publication time distribution, i.e. distribution of the documents' timestamps

Collection Features

(2) Distribution of timestamps

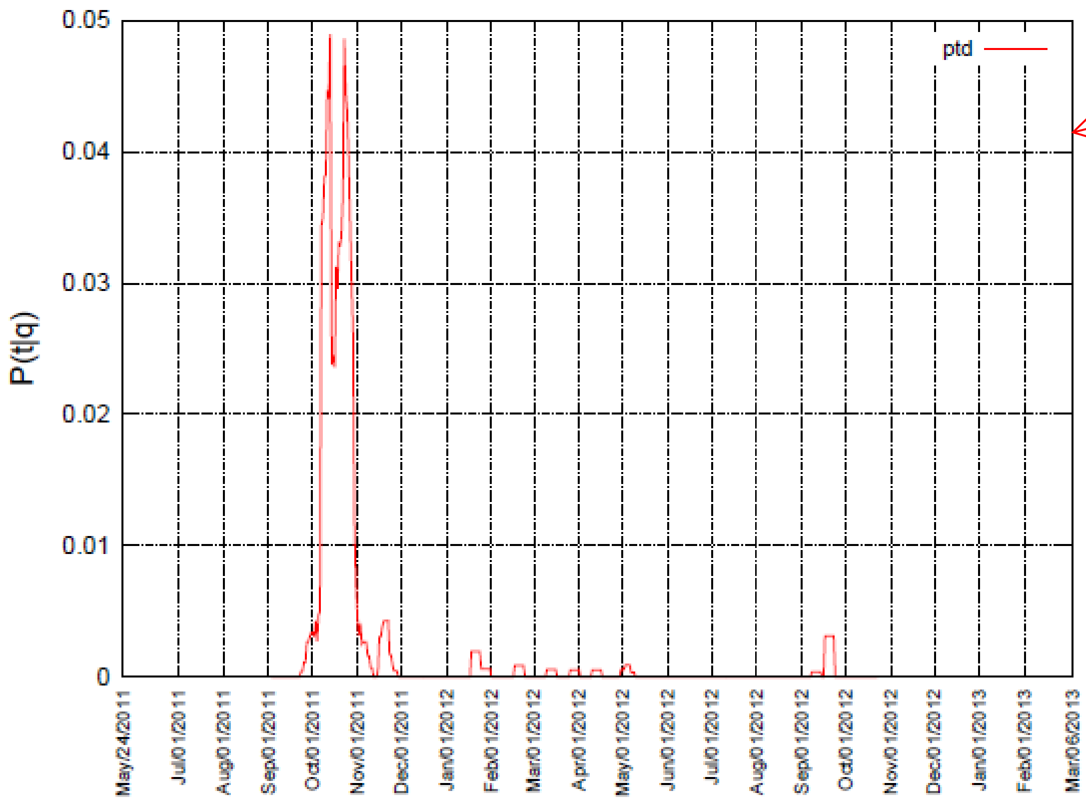

Figure 4.3.: PTD for Query "occupy wall street"

Publication time distribution, i.e. distribution of the documents' timestamps

Example Features:

- How „peaky" is the curve? (**kurtosis**)
- How much does the distribution deviate from the background distribution? (**temporal KL divergence**)

Collection Features

(3) Temporal Dictionary

Iterate over the complete document collection and for each unigram/bigram *x* create a dictionary entry containing the following scores:

- **dictPastScore**
- **dictRecencyScore**
- **dictFutureScore**
- **avgTimex**

„fraction of sentences with x that contain a temporal expression which refers to the past/present/future“ (relative to document timestamp)

„average number of temporal expressions that appear together with x“

Collection Features

(3) Temporal Dictionary

Example:

	„live stream“	„moon landing“
dictPastScore	0.08	**0.28**
dictRecencyScore	**0.20**	0.08
dictFutureScore	0.07	0.09
avgTimex	0.49	0.75

Given a query, we can now determine average values of the query terms as features! (Simple dictionary lookup)

Linguistic Features

(any features obtained from NLP taggers)

Linguistic Features

(1) POS/Named Entity Features (Examples)

- **startsWithNoun:** „time in london“
- **startsWithUnpersonalVerb:** „lose weight quickly“
- **startsWithInflectedVerb:** „did the Pirates win today“
- **containsEntity:** „nba playoffs 2013 standings“

No obvious relation to temporal class!

Linguistic Features

(2) Tense Features

- **containsPastTense:** „what was the cold war“
- **containsPresentTense:** „time is of the essence“
- **containsFutureTense:** „when will the sun rise tomorrow“

Linguistic Features

(3) Temporal expressions in the query string

- **containsPastDate:** „fifa world cup 2006“
- **containsFutureDate:** „fifa world cup 2018“
- Different features for temporal expressions which refer to a recent date

Overview

1. Feature Design
2. Results
3. Conclusion

Setup

We train two classifiers on our training set (using the previously designed features):

- Naive Bayes classifier (WEKA implementation **NaiveBayes**)
- Decision tree (WEKA implementation **J48**)

We use these classifiers to assign classes to unseen instances.

Setup

We use two baselines:

- **Random classifier : 25 %**
- **Unigram/bigram feature classifier : 56.30 %**

We can correctly predict the class of more than half of our instances only based on unigrams and bigrams!

Results

Accuracy values from the formal runs (in %):

	total	P	R	F	A
Run 1:	**62.33**	**53**	**57**	**65**	**73**
Run 2:	**64.00**	**60**	**49**	**71**	**76**
Run 3:	**61.67**	**60**	**44**	**63**	**80**

Our approach yields a relative improvement of ~10% over the baseline.

Results

Insights:

- Temporal dictionary: we can correctly classify over 50% of our queries with only three features!
- Linguistic features: exploiting linguistic query properties could be useful (on queries with certain properties)

Maybe not so effective in the real world…

Overview

1. Feature Design
2. Results
3. Conclusion

Conclusion

This presentation

- gave a concise overview over some features that we designed
  - Collection features: time distributions, temporal dictionary…
  - Linguistic features: POS, NER, tense, temporal expressions
- summarized the experimental results
  - Accuracy estimates from formal runs
  - Observations

Thank you for your attention!

References

Temporal Query Classification, Bachelor‘s Thesis, Robin Burghartz, 2014.