Internship Report: Timeline: Cross Document Event Ordering

Swati Gupta • 16.07.2017

Inspiration

Inspiration:

My work during the internship is inspired by the TimeLine task that was organised within the time and space track of SemEval 2015.

After careful consideration of the various possibilities with my supervisor, I chose to work on this task, as it seemed the most interesting and relevant to current scenerio.

The goal of the project is to reconstruct story lines across news articles in order to provide policy and decision makers with an overview of what happened, to whom,when, and where. Thus, aiming to present end-users with cross-document storylines.

Task Description

Problem Statement:

Given a set of documents and a set of target entities, the task consists of building an ordered timeline for each entity, by: A) Detecting, B) anchoring in time and C) ordering the events involving that entity.

In my focus of work (Track B of the SemEval task): the gold level event annotations have been provided. Thus, (A) simplifies to resolving the events relevant to the entity among all events provided, and not a full fledged detection of those events in the raw data.

Key Words

‘EVENT’

- The notion of event is based on TimeML, according to which an event is a cover term for situations that happen or occur, including predicates describing states or circumstances in which something obtains or holds true

‘TARGET ENTITY’

- Target entities explicitly participate in a has_participant relation with the semantic role ARG0 (i.e. agent)or ARG1 (i.e. patient), with the events.
- Ex:

  He has been **fighting** pancreatic cancer since 2004.

  The current Microsoft CEO **described** Jobs as “one of the founders of our industry and a true visionary

Details

Input

- A set of documents and A set of target entities; only entities involved in more than two events across at least two different documents are considered as candidates target entities.
- For track B, gold event mentions are also available.
- For training, the gold annotated timelines for each target entity is also available.

Expected End Result

- The expected output is one timeline for each target entity. A timeline for a specific target entity consists of the ordered list of the events in which that entity participates. Events in a timeline are anchored in time through the time anchor attribute.
- Format:

  1 2011-10-06 18355-18-described

Initial Approach

- Browsed the internet for relevant information, and summarised all the papers written by the participating Teams as well as others who had tackled the problem.
- To kick start, I focused on a single document and a single target entity, and lookout for possible heuristics to create a rough timeline.
- Tested Tools like NLTK and Stanford CoreNLP and compared their performance. (Decided in favour of coreNLP)

Stanford CoreNLP:

An integrated NLP toolkit with a good range of grammatical analysis tools. It can perform tokenization, sspliting, part of speech tagging, lemma and named entity recognition, parsing and coreference resolution.

Steps taken

- Matching: Identify sentences in the document in which parts of target entity name occur
- Coreference Resolution: extract sentences with implicit mentions of the entity as opposed to ones that only contain explicit mentions
- Temporal Tagging: detect time mentions and resolve them to the TIMEX date format
- Linking event to time mentions : an open problem (Ref later slides)
- Lemma Clustering: Cluster two entries if they correspond to same event occurrence.
- Ordering on the timeline, based on occurrence dates.

After Lemma Clustering

```
swati@swati-Lenovo-Z50-70 ~/Documents/intern
wler_v2.py
['2011-01-17'] 17677-2 announced
['2011-01-17'] 17677-2 leave
['2011-01-17'] 17677-4 announcement
['2011-01-17'] 17677-4 marked
['2011-01-17'] 17677-4 leave
['2009', '2004'] 17677-5 fought
['2009', '2004'] 17677-5 cancer
['2009', '2004'] 17677-5 revealing
['2009', '2004'] 17677-5 undergone
['2009', '2004'] 17677-5 transplant
['2011-01-17'] 17677-6 decision
['2011-01-17'] 17677-6 said
['2011-01-17'] 17677-7 left
['2011-01-17'] 17677-7 operations
['2011-01-17'] 17677-7 move
['2011-01-17'] 17677-7 absences
['2011-01-17'] 17677-8 reported
['2011-01-17'] 17677-8 situation
['2011-01-17'] 17677-8 said
['2011-01-17'] 17677-8 problems
['2011-01-17'] 17677-8 problems
['2011-01-17'] 17677-8 experienced
['2011-01-17'] 17677-8 transplants
['2011-01-17'] 17677-9 asked
['2011-01-17'] 17677-9 said
['2011-01-17'] 17677-11 released
['2011-01-17'] 17677-11 reads
['2011-01-17'] 17677-11 request
['2011-01-17'] 17677-11 granted
['2011-01-17'] 17677-11 leave
['2011-01-17'] 17677-12 involved
['2011-01-17'] 17677-12 decisions
swati@swati-Lenovo-Z50-70 ~/Documents/intern
```

```
[['2009', '2004'], '17677-5', 'cancer']
[['2011-01-17'], '17677-7', 'move']
[['2011-01-17'], '17677-12', 'involved']
[['2011-01-17'], '17677-8', 'reported']
[['2011-01-17'], '17677-6', 'said', ['2011-01-17'], '17677-8', 'said', ['2011-01
-17'], '17677-9', 'said']
[['2011-01-17'], '17677-7', 'operations']
[['2011-01-17'], '17677-4', 'marked']
[['2009', '2004'], '17677-5', 'transplant', ['2011-01-17'], '17677-8', 'transpla
nts']
[['2011-01-17'], '17677-4', 'announcement']
[['2011-01-17'], '17677-11', 'granted']
[['2011-01-17'], '17677-6', 'decision', ['2011-01-17'], '17677-12', 'decisions']
[['2009', '2004'], '17677-5', 'fought']
[['2011-01-17'], '17677-7', 'absences']
[['2011-01-17'], '17677-2', 'announced']
[['2011-01-17'], '17677-11', 'reads']
[['2011-01-17'], '17677-9', 'asked']
[['2009', '2004'], '17677-5', 'revealing']
[['2011-01-17'], '17677-11', 'request']
[['2011-01-17'], '17677-8', 'experienced']
[['2011-01-17'], '17677-2', 'leave', ['2011-01-17'], '17677-4', 'leave', ['2011-
01-17'], '17677-7', 'left', ['2011-01-17'], '17677-11', 'leave']
[['2011-01-17'], '17677-11', 'released']
[['2011-01-17'], '17677-8', 'situation']
[['2011-01-17'], '17677-8', 'problems', ['2011-01-17'], '17677-8', 'problems']
[['2009', '2004'], '17677-5', 'undergone']
```

Problems Faced

Clearly, the linking of events with time mentions was a major source of error, as I could not resolve whenever there was a possibility of two different times being linked to the same event.

Ex: In a sentence like this:
He <EVENT>fought</EVENT> pancreatic cancer in 2004 and <EVENT>took</EVENT> six months off in early 2009.

In order to resolve this, I turned to researching what others had done. The winning team at SemEval 2015 had made use of a tool called TIPSem.

TIPSem

TIPSem is a temporal information extraction software which automatically annotates from raw text - TimeML temporal expressions, events, and temporal relations (tlink).
This was the tool used by the winning team in the semEval Task.

I installed and tested this tool, to see how well it was able to resolve links in cases of ambiguity. While, it performed reasonably well otherwise, it often returned wrong results.

Ex: In the sentence mentioned:
He <EVENT>fought</EVENT> pancreatic cancer in 2004 and <EVENT>took</EVENT> six months off in early 2009.

TIPSem linked ‘fought’ with document creation time, and ‘took’ with 2004, which is clearly wrong.

Machine Learning to the rescue

After discussions with my supervisor, I decided to build an SVM Classifier to link entity and time mentions. Since an inherent and intuitive way to visualise a link and extract a relation is through trees, I used tree kernels (like the Subset Tree Kernel, SubTree Kernel etc) and tree structures like Dependency Trees (as Grammatical Relations and words/lemmas tree), Part of speech tagging Trees along with some flat features(like path length and TIPSem output) to train the svm classifier.

After testing on various configurations and parameter values, I obtained good results, but with a lot of scope for improvement and further work. (Ref next slide)

Results for a particular configuration:

1)I used the SST kernel for training.(the sequential summation of trees-
using SST kernel, is summed to the sequential summation of vectors using a
polynomial kernel with degree = 5.)
2)The training data, for each +ve or -ve example consisted of two trees:
the dependency tree(GRW), Pos tree,and three features: shortest path
length in DT shortest path length in POS tree,
TIPSEM output(+1 if tipsem
gave a link, -1 if it did not).
3)The +ve or -ve ness of the example is decided by the fact whether the link
appears in the gold level timeline for the target entity or not.

4) At this time, I only used a single target entity: 'steve jobs'. Out of
the 17 articles on steve jobs, 14 were used to create the training set,
and 3 were used to create testing set.

RESULTS:

Accuracy on test set: **89.04%** (65 correct, 8 incorrect, 73 total)
Precision/recall on test set: **100.00%/55.56%**
F1-measure on test set: **71.43%** (Snapshot on the next slide)

```
bash-4.2$ ./svm_learn -t 5 -C + -F 1 -S 1 -d 5 training_data.txt model_file
Scanning examples...done
Reading examples into memory...100..OK. (174 examples read)

Number of examples: 174, linear space size: 3

estimating ...
Setting default regularization parameter C=0.3333
Optimizing.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................
 Checking optimality of inactive variables...done.
 Number of inactive variables = 107
done. (492 iterations)
Optimization finished (17 misclassified, maxdiff=0.00099).
Runtime in cpu-seconds: 0.04
Number of SV: 86 (including 45 at upper bound)
L1 loss: loss=36.88017
Norm of weight vector: |w|=2.63862
Norm of longest example vector: |x|=1.73205
Estimated VCdim of classifier: VCdim<=21.88695
Computing XiAlpha-estimates...done
Runtime for XiAlpha-estimates in cpu-seconds: 0.00
XiAlpha-estimate of the error: error<=25.86% (rho=1.00,depth=0)
XiAlpha-estimate of the recall: recall=>6.67% (rho=1.00,depth=0)
XiAlpha-estimate of the precision: precision=>10.53% (rho=1.00,depth=0)
Number of kernel evaluations: 15508
Writing model file...done
bash-4.2$ ./svm_classify -v 3 testing_data.txt model_file
Reading model...OK. (86 support vectors read)
Classifying test examples..done
Runtime (without IO) in cpu-seconds: 0.01
Accuracy on test set: 89.04% (65 correct, 8 incorrect, 73 total)
Precision/recall on test set: 100.00%/55.56%
F1-measure on test set: 71.43%
```

Next steps

1) **Incorporate the whole Dataset and all target entities**
2) **Create a timeline out of the output from the classifier**
3) **Test on the official dataset, and compare the results.**
4) **Improve upon and add to the flat features for the trainer.**

I plan to keep working on the problem after I go back, and keep my supervisor updated on the progress.

Acknowledgement

1. I would like to sincerely thank the Computer Science Department at UT Dallas for supporting me and giving me the opportunity to come here and work on this project

2. I would especially like to thank Prof. Gupta and Prof. Karrah for taking such good care of us, and Prof. Ng for guiding and supervising me throughout the project. This would not have been possible without his constant encouragement.