A comparison and analysis of models for event trigger detection

Shang Chun Sam Wei
School of Information Technologies
University of Sydney
NSW 2006, Australia
swei4829@uni.sydney.edu.au

Ben Hachey
School of Information Technologies
University of Sydney
NSW 2006, Australia
ben.hachey@gmail.com

Abstract

Interpreting event mentions in text is central to many tasks from scientific research to intelligence gathering. We present an event trigger detection system and explore baseline configurations. Specifically, we test whether it is better to use a single multi-class classifier or separate binary classifiers for each label. The results suggest that binary SVM classifiers outperform multi-class maximum entropy by 6.4 points F-score. Brown cluster and WordNet features are complementary with more improvement from WordNet features.

1 Introduction

Events are frequently discussed in text, e.g., criminal activities such as violent attacks reported in police reports, corporate activities such as mergers reported in business news, biological processes such as protein interactions reported in scientific research. Interpreting these mentions is central to tasks like intelligence gathering and scientific research. Event extraction automatically identifies the triggers and arguments that constitute a textual mention of an event in the world. Consider:

Bob **bought** *the book from Alice.*

Here, a trigger – "bought" (*Transaction.Transfer–Ownership*) – predicates an interaction between the arguments – "Bob" (*Recipient*), "the book" (*Thing*) and "Alice" (*Giver*). We focus on the trigger detection task, which is the first step in event detection and integration.

Many event extraction systems use a pipelined approach, comprising a binary classifier to detect event triggers followed by a separate multi-class classifier to label the type of event (Ahn, 2006). Our work is different in that we use a single classification step with sub-sampling to handle data skew. Chen and Ji (2009) use Maximum Entropy (ME) classifier in their work. However, their approach is similar to (Ahn, 2006) where they identify the trigger first then classify the trigger at later stage. Kolya et al. (2011) employ a hybrid approach by using Support Vector Machine (SVM) classifier and heuristics for event extraction.

We present an event trigger detection system that formulates the problem as a token-level classification task. Features include lexical and syntactic information from the current token and surrounding context. Features also include additional word class information from Brown clusters, WordNet and Nomlex to help generalise from a fairly small training set. Experiments explore whether multi-class or binary classification is better using SVM and ME.

Contributions include: (1) A comparison of binary and multi-class versions of SVM and ME on the trigger detection task. Experimental results suggest binary SVM outperform other approaches. (2) Analysis showing that Brown cluster, Nomlex and WordNet features contribute nearly 10 points F-score; WordNet+Nomlex features contribute more than Brown cluster features; and improvements from these sources of word class information increase recall substantially, sometimes at the cost of precision.

2 Event Trigger Detection Task

We investigate the event trigger detection task from the 2015 Text Analysis Conference (TAC) shared task (Mitamura and Hovy, 2015). The task defines 9 event types and 38 subtypes such as *Life.Die, Conflict.Attack, Contact.Meet*. An event trigger is the smallest extent of text (usually a word or short phrase) that predicates the occurrence of an event (LDC, 2015).

In the following example, the words in bold trigger *Life.Die* and *Life.Injure* events respectively:

Sam Shang Chun Wei and Ben Hachey. 2015. A comparison and analysis of models for event trigger detection . In *Proceedings of Australasian Language Technology Association Workshop*, pages 128−132.

The explosion **killed** *7 and* **injured** *20.*

Note that an event mention can contain multiple events. Further, an event trigger can have multiple events. Consider:

The **murder** *of John.*

where "murder" is the trigger for both a *Conflict.Attack* event and a *Life.Die* event. Table 1 shows the distribution of the event subtypes in the training and development datasets.

3 Approach

We formulate event trigger detection as a token-level classification task. Features include lexical and semantic information from the current token and surrounding context. Classifiers include binary and multi-class versions of SVM and ME.

As triggers can be a phrase, we experimented with Inside Outside Begin 1 (IOB1) and Inside Outside Begin 2 (IOB2) encodings (Sang and Veenstra, 1999). Table 2 contains an example illustrating the two schemes. Preliminary results showed little impact on accuracy. However, one of the issues with this task is data sparsity. Some event subtypes have few observations in the corpus. IOB2 encoding increases the total number of categories for the dataset. Thus make the data sparsity issue worse. Therefore we use the IOB1 encoding for the rest of the experiments.

Another challenge is that the data is highly unbalanced. Most of the tokens are not event triggers. To address this, we various subsets of negative observations. Randomly sampling 10% of the negative examples for training works well here.

3.1 Features

All models used same rich feature sets. The features are divided into three different groups.

Feature set 1 (FS1): Basic features including following. (1) Current token: Lemma, POS, named entity type, is it a capitalised word. (2) Within the window of size two: unigrams/bigrams of lemma, POS, and name entity type. (3) Dependency: governor/dependent type, governor/dependent type + lemma, governor/dependent type + POS, and governor/dependent type + named entity type.

Feature set 2 (FS2): Brown cluster trained on the Reuters corpus (Brown et al., 1992; Turian et al., 2010) with prefix of length 11, 13 and 16.[1]

Feature set 3 (FS3): (1) WordNet features including hypernyms and synonyms of the current token. (2) Base form of the current token extracted from Nomlex (Macleod et al., 1998).[2]

Event Subtype	Train	Dev
Business.Declare-Bankruptcy	30	3
Business.End-Org	11	2
Business.Merge-Org	28	0
Business.Start-Org	17	1
Conflict.Attack	541	253
Conflict.Demonstrate	162	38
Contact.Broadcast	304	112
Contact.Contact	260	77
Contact.Correspondence	77	18
Contact.Meet	221	23
Justice.Acquit	27	3
Justice.Appeal	25	12
Justice.Arrest-Jail	207	79
Justice.Charge-Indict	149	41
Justice.Convict	173	49
Justice.Execute	51	15
Justice.Extradite	62	1
Justice.Fine	53	2
Justice.Pardon	221	18
Justice.Release-Parole	45	28
Justice.Sentence	118	26
Justice.Sue	54	1
Justice.Trial-Hearing	172	24
Life.Be-Born	13	6
Life.Die	356	157
Life.Divorce	45	0
Life.Injure	63	70
Life.Marry	60	16
Manufacture.Artifact	18	4
Movement.Transport-Artifact	52	18
Movement.Transport-Person	390	125
Personnel.Elect	81	16
Personnel.End-Position	130	79
Personnel.Nominate	30	5
Personnel.Start-Position	60	17
Transaction.Transaction	34	17
Transaction.Transfer-Money	366	185
Transaction.Transfer-Ownership	233	46

Table 1: Event subtype distribution.

[1] http://metaoptimize.com/projects/wordreprs/

[2] http://nlp.cs.nyu.edu/nomlex/

Word	IOB1	IOB2
He	O	O
has	O	O
been	O	O
found	I-Justice.Convict	B-Justice.Convict
guilty	I-Justice.Convict	I-Justice.Convict
for	O	O
the	O	O
murder	I-Life.Die	B-Life.Die
.	O	O

Table 2: IOB1 and IOB2 encoding comparison. "B" represents the first token of an event trigger. "I" represents a subsequent token of a multi-word trigger. "O" represents no event.

3.2 Classifiers

We train multi-class ME and SVM classifiers to detect and label events. L-BFGS (Liu and Nocedal, 1989) is used as the solver for ME. The SVM uses a linear kernel. We also compare binary versions of ME and SVM by building a single classifier for each event subtype.

4 Experimental setup

4.1 Dataset

The TAC 2015 training dataset (LDC2015E73) is used for the experiment. The corpus has a total of 158 documents from two genres: 81 newswire documents and 77 discussion forum documents. Preprocessing includes tokenisation, sentence splitting, POS tagging, named entity recognition, constituency parsing and dependency parsing using Stanford CoreNLP 3.5.2.[3]

The dataset is split into 80% for training (126 documents) and 20% for development (32 documents. Listed in Appendix A).

4.2 Evaluation metric

Accuracy is measured using the TAC 2015 scorer.[4] Precision, recall and F-score are defined as:

$$P = \frac{TP}{N_S}; R = \frac{TP}{N_G}; F1 = \frac{2PR}{P+R}$$

where TP is the number of correct triggers (true positives), N_S is the total number of predicted system mentions, and N_G is the total number of annotated gold mentions. An event trigger is counted as correct only if the boundary, the event type and the event subtype are all correctly identified. We report micro-averaged results.

[3] http://nlp.stanford.edu/software/corenlp.shtml

[4] http://hunterhector.github.io/EvmEval/

5 Results

Table 3 shows the results from each classifier. The binary SVMs outperform all other models with an F-score of 55.7. The score for multi-class SVM is two points lower at 53.2. Multi-class and binary ME comes next with binary performing worst.

System	P	R	F1
Multi-class ME	62.2	40.8	49.2
Multi-class SVM	55.6	50.9	53.2
Binary ME	77.8	28.2	41.4
Binary SVM	64.7	48.9	**55.7**

Table 3: System performance comparison.

5.1 Feature set

We perform a cumulative analysis to quantify the contribution of different feature sets. Table 4 shows that feature set 2 (Brown cluster) helped with recall sometimes at the cost of precision. The recall is further boosted by feature set 3 (WordNet and Nomlex). However, the precision dropped noticeably for SVM models.

System	P	R	F1
Multi-class systems			
ME FS1	54.1	16.9	25.8
ME FS1+FS2	57.8	21.3	31.1
ME FS1+FS2+FS3	62.2	40.8	49.2
SVM FS1	62.1	35.3	45.0
SVM FS1+FS2	60.9	39.3	47.8
SVM FS1+FS2+FS3	55.6	50.9	53.2
Binary systems			
ME FS1	64.7	6.1	11.2
ME FS1+FS2	72.7	10.1	17.8
ME FS1+FS2+FS3	77.8	28.2	41.4
SVM FS1	71.0	34.2	46.2
SVM FS1+FS2	70.5	38.4	49.7
SVM FS1+FS2+FS3	64.7	48.9	**55.7**

Table 4: Feature sets comparison.

5.2 Performance by event subtype

Figure 1 shows how classifiers perform on each event subtype. Binary SVM generally has better recall and slightly lower precision. Hence, the overall performance of the model improves.

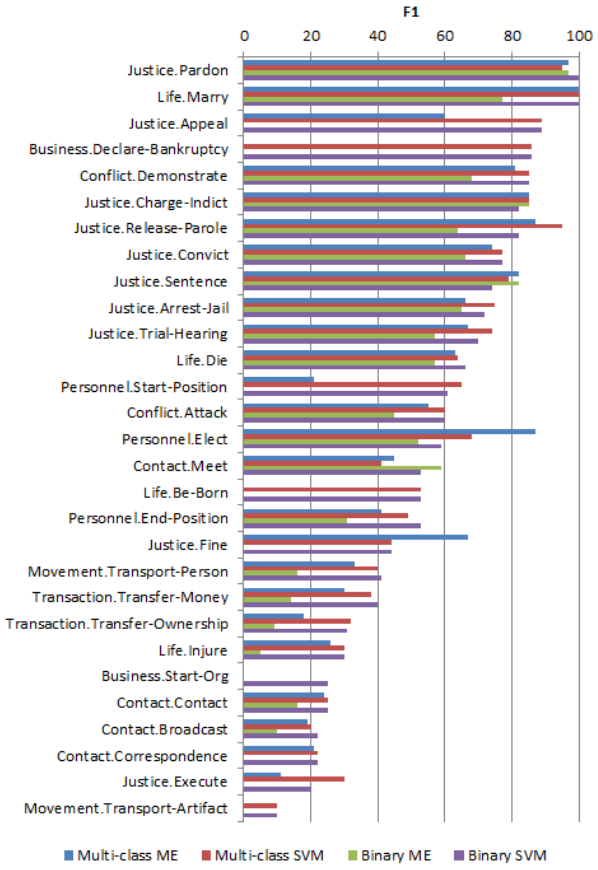

Figure 1: Performance by subtype.

5.3 Error analysis

We sampled 20 precision and twenty recall errors from the binary SVM classifier. 40% of precision errors require better modelling of grammatical relations, e.g., labelling "focus has moved" as a transport event. 35% require better use of POS information, e.g., labelling "said crime" as a contact event. 65% of recall errors are tokens in multiword phrases, e.g., "going to jail". 45% are triggers that likely weren't seen in training and require better generalisation strategies. Several precision and recall errors seem to actually be correct.

6 Conclusion

We presented an exploration of TAC event trigger detection and labelling, comparing classifiers and rich features. Results suggest that SVM outperforms maximum entropy and binary SVM gives the best results. Brown cluster information increases recall for all models, but sometimes at the cost of precision. WordNet and Nomlex features provide a bigger boost, improving F-score by 6 points for the best classifier.

Acknowledgements

Sam Wei is funded by the Capital Markets Cooperative Research Centre. Ben Hachey is the recipient of an Australian Research Council Discovery Early Career Researcher Award (DE120102900)

References

David Ahn. 2006. The stages of event extraction. In *COLING-ACL Workshop on Annotating and Reasoning About Time and Events*, pages 1–8.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Zheng Chen and Heng Ji. 2009. Language specific issue and feature exploration in chinese event extraction. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–212.

Anup Kumar Kolya, Asif Ekbal, and Sivaji Bandyopadhyay. 2011. A hybrid approach for event extraction and event actor identification. In *International Conference on Recent Advances in Natural Language Processing*, pages 592–597.

LDC, 2015. *Rich ERE Annotation Guidelines Overview*. Linguistic Data Consortium. Version 4.1. Accessed 14 November 2015 from `http://cairo.lti.cs.cmu.edu/kbp/2015/event/summary_rich_ere_v4.1.pdf`.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528.

Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. 1998. Nomlex: A lexicon of nominalizations. In *Euralex International Congress*, pages 187–193.

Teruko Mitamura and Eduard Hovy, 2015. *TAC KBP Event Detection and Coreference Tasks for English*. Version 1.0. Accessed 14 November 2015 from `http://cairo.lti.cs.cmu.edu/kbp/2015/event/Event_Mention_Detection_and_Coreference-2015-v1.1.pdf`.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 173–179.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semisupervised learning. In *Annual Meeting of the Association for Computational Linguistics*, pages 384–394.

Appendix A: Development set document IDs

3288ddfcb46d1934ad453afd8a4e292f
AFP_ENG_20091015.0364
AFP_ENG_20100130.0284
AFP_ENG_20100423.0583
AFP_ENG_20100505.0537
AFP_ENG_20100630.0660
APW_ENG_20090605.0323
APW_ENG_20090611.0337
APW_ENG_20100508.0084
APW_ENG_20101214.0097
CNA_ENG_20101001.0032
NYT_ENG_20130628.0102
XIN_ENG_20100114.0378
XIN_ENG_20100206.0090
bolt-eng-DF-170-181103-8901874
bolt-eng-DF-170-181103-8908896
bolt-eng-DF-170-181109-48534
bolt-eng-DF-170-181109-60453
bolt-eng-DF-170-181118-8874957
bolt-eng-DF-170-181122-8791540
bolt-eng-DF-170-181122-8793828
bolt-eng-DF-170-181122-8803193
bolt-eng-DF-199-192783-6864512
bolt-eng-DF-199-192909-6666973
bolt-eng-DF-200-192403-6250142
bolt-eng-DF-200-192446-3810246
bolt-eng-DF-200-192446-3810611
bolt-eng-DF-200-192451-5802600
bolt-eng-DF-200-192453-5806585
bolt-eng-DF-203-185933-21070100
bolt-eng-DF-203-185938-398283
bolt-eng-DF-212-191665-3129265