Improving Event Detection with Abstract Meaning Representation

Xiang Li Thien Huu Nguyen Kai Cao Ralph Grishman
Computer Science Department
New York University
New York, NY 10003, USA
{xiangli,thien,kcao,grishman}@cs.nyu.edu

Abstract

Event Detection (ED) aims to identify instances of specified types of events in text, which is a crucial component in the overall task of event extraction. The commonly used features consist of lexical, syntactic, and entity information, but the knowledge encoded in the Abstract Meaning Representation (AMR) has not been utilized in this task. AMR is a semantic formalism in which the meaning of a sentence is encoded as a rooted, directed, acyclic graph. In this paper, we demonstrate the effectiveness of AMR to capture and represent the deeper semantic contexts of the trigger words in this task. Experimental results further show that adding AMR features on top of the traditional features can achieve 67.8% (with 2.1% absolute improvement) F-measure (F_1), which is comparable to the state-of-the-art approaches.

1 Introduction

The problem of event detection (ED) is identifying instances of specified types of events in text. Associated with each event mention, the event trigger (most often a single verb or nominalization) evokes that event. Our task, more precisely stated, involves identifying event triggers and classifying them into specific types. In this paper, we focus on the event detection task defined in Automatic Content Extraction (ACE) evaluation[1]. The task defines 8 event types and 33 subtypes such as *Die* and *End-Position*. For instance, according to the ACE 2005 annotation guideline, in the sentence "*A bomb **exploded** in central Baghdad yesterday*", an event detection system should be able to recognize the word "*exploded*" as a trigger for the event *Attack*. ED is a crucial component in the overall task of event extraction, which also involves event argument discovery[2]. This task is quite challenging, as the same event might appear with various trigger expressions, and an expression might also represent different events in different contexts.

Abstract Meaning Representation (AMR) (Dorr et al., 1998; Banarescu et al., 2013) (§2) is a semantic formalism in which the meaning of a sentence is encoded as a rooted, directed, acyclic graph. Nodes represent concepts, and labeled directed edges represent the relationships between them. The knowledge incorporated in the AMR (§3) can benefit the ED task by abstracting the semantic representation from the sentences with the same meaning but possibly in different syntactic forms. The results demonstrate that some characteristics are not completely captured by traditional features (e.g., dependency parse features), but may be revealed in the AMR, complementing other features to help boost the performance to 67.8% (with 2.1% absolute improvement) in F_1 (§4).

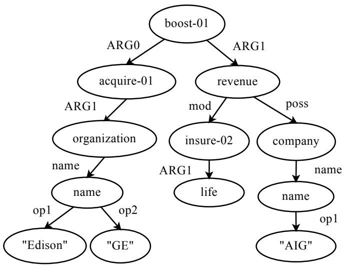
2 Abstract Meaning Representation

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is a sembanking language that captures whole sentence meanings in a rooted, directed, labeled, and (predominantly) acyclic graph structure - see Figure 1 for an example AMR parse. AMR utilizes multi-layer linguistic analysis such as PropBank frames, non-core semantic roles, coreference, named entity annotation, modality and negation to represent the semantic structure of a sentence. AMR strives for a more logical, less syntactic representation, collapsing some word category (verbs and nouns), word order, and morphological variation. Instead, it focuses on semantic relations between concepts and makes heavy use of predicate-argument structures as defined in PropBank (Kingsbury and Palmer,

[1] http://projects.ldc.upenn.edu/ace/

[2] Argument identification and argument role labeling are out of the scope of this paper, as planned for the future work.

Proceedings of the First Workshop on Computing News Storylines, pages 11–15,
Beijing, China, July 31, 2015. ©2015 Association for Computational Linguistics and The Asian Federation of Natural Language Processing



(a) AMR graph

```
b / boost-01
 :ARG0  a / acquire-01
   :ARG1  o / organization
     :name  n2 / name
       :op1   dison
       :op2  G
 :ARG1  r / revenue
   :mod  i / insure-02
     :ARG1  l / life
   :poss  c / company
     :name  n / name
       :op1  A G
```

(b) AMR annotation

Figure 1: Two equivalent ways of representing the AMR parse for the example sentence, "*The acquisition of Edison GE will boost AIG's annual life insurance revenue.*"

2002; Palmer et al., 2005). For example, a phrase like "*bond investor*" is represented using the frame "`invest-01`", even though no verbs appear.

In addition, many function words (determiners, prepositions) are considered to be syntactic "sugar" and are not explicitly represented in AMR, except for the semantic relations they signal. Hence, it assigns the same AMR parse graph to sentences that have the same basic meaning.[3]

Compared to traditional dependency parsing and semantic role labeling, the nodes in AMR are entities instead of words, and the edge types are much more fine-grained. AMR thus captures deeper meaning compared with other representations which are more commonly used to represent context in ED. In this work, all AMR parse graphs are automatically generated from the first published AMR parser, *JAMR* (Flanigan et al., 2014).

3 Framework and Features

To compare our proposed AMR features with the previous approaches, we implemented a Maximum Entropy (MaxEnt) classifier with both traditional features and AMR features for trigger identification and label classification.

To make a fair comparison, the feature sets in the baseline are identical to the local text features in (Li et al., 2013b). From Table 2, we can see that this baseline MaxEnt classifier with local features aligns well with the joint beam search approach using perceptron and local features in (Li et al., 2013b). The slight variation is mainly due to the different pre-processing procedures for features.

On top of the local features used in the baseline MaxEnt classifier, we exploit knowledge from AMR parse graphs to add AMR features into the MaxEnt classifier. The effects of these features have been explored based on the performance on the development dataset. More features have actually been studied, such as the features extracted from the grandparent node, the conjunction features of candidate and parent nodes, etc. Table 1 lists the final AMR features extracted from the AMR parse graph, and the corresponding feature values, for trigger candidate "*acquisition*", from the above example AMR graph.

4 Experiments

In this section, we will compare our MaxEnt classifiers using both baseline features and additional proposed AMR features with the state-of-the-art systems on the blind test set, and then discuss the results in more detail.

4.1 Dataset and Evaluation Metric

We evaluate our system with above presented features over the ACE 2005 corpus. For comparison purposes, we utilize the same test set with 40 newswire articles (672 sentences), the same development set with 30 other documents (836 sentences) and the same training set with the remaining 529 documents ($14,849$ sentences) as the previous studies on this dataset (Ji and Grishman, 2008; Liao and Grishman, 2010; Li et al., 2013b).

Following the previous work (Ji and Grishman, 2008; Liao and Grishman, 2010; Hong et al., 2011; Li et al., 2013b), a trigger candidate is counted as correct if its event subtype and offsets match those of a reference trigger. The ACE 2005 corpus has 33 event subtypes that, along with one class "*Other*" for the non-trigger tokens, constitutes a 34-class classification problem in this work. Finally we use *Precision (P)*, *Recall (R)*, and *F-measure (F_1)* to evaluate the performance. Table 2 presents the overall performance of the systems with gold-standard entity mention and

[3] Readers can refer to (Banarescu et al., 2013) for a complete description of AMR and more examples.

Node	Feature	Description	Example
Candidate	amr_word_tag	The conjunction of the candidate word and its AMR tag	`acquire-01_ARG0`
Root	amr_dist_to_root	The distance between the candidate word and the root	`1`
Parent	amr_parent_word	The word of the parent node	`boost-01`
	amr_parent_tag	The AMR tag of the parent node	`AMR-Root`
	amr_parent_word_tag	The conjunction of the parent word and its AMR tag	`boost-01_AMR-Root`
Sibling	amr_sibling_tag	The AMR tag of each sibling node	`ARG1`
	amr_sibling_word_tag	The conjunction of the sibling word and its AMR tag	`revenue_ARG1`
Children	amr_child_word_tag	The conjunction of the child word and its AMR tag	`organization_ARG1`
Grandchildren	amr_grandchild_word	The word of the grandchild node	`name`

Table 1: Features extracted from the AMR graph and example features for candidate "*acquisition*".

Methods	P	R	F_1
Sentence-level in Hong et al. (2011)	67.6	53.5	59.7
MaxEnt classifier with local features in Li et al. (2013b)	74.5	59.1	65.9
Joint beam search with local features in Li et al. (2013b)	73.7	59.3	65.7
Joint beam search with local and global features in Li et al. (2013b)	73.7	62.3	67.5
Cross-entity in Hong et al. (2011) †	72.9	64.3	68.3
MaxEnt classifier with baseline features	70.8	61.4	65.7
MaxEnt classifier with baseline + AMR features	74.4	62.3	**67.8**

Table 2: Performance (%) comparison with the state-of-the-art systems. † beyond sentence level.

type information[4].

As we can see from Table 2, among the systems that only use sentence level information, our MaxEnt classifier using both baseline and AMR features significantly outperforms the MaxEnt classifier with baseline features as well as the joint beam search with local features from Li et al. (2013b) (an absolute improvement of 2.1% in F_1 score), and performs comparably (67.8% in F_1) to the state-of-the-art joint beam search approach using both local and global features (67.5% in F_1) (Li et al., 2013b). This is remarkable since our MaxEnt classifier does not require any global features[5] or sophisticated machine learning framework with a much larger hypothesis space, e.g., structured perceptron with beam search (Li et al., 2013b).

From the detailed result analysis, we can see that the event trigger detection of most event types are significantly ($p < 0.05$) improved over the baseline setting. Many types gain substantially in both precision and recall, while only 4 out of 33 event types decrease slightly in performance. Table 3 presents the performance comparison for a subset of event types between the baseline and the classifier with both baseline and AMR features[6].

For instance, in the test sentence "*. . . have Scud missiles capable of* **reaching** *Israel . . .* ", the trigger candidate "*reach*" can be a *Conflict:Attack* event (as in this case) but also a *Contact:Phone-Write* event (e.g., "*they tried to reach their loved ones*"). If the subject (`ARG0`) is a weapon (as in this example), it should be an *Attack* event. This pattern can be learned from a sentence such as "*The missiles . . . reach their target*". The AMR parser is able to look through "*capable of*" and recognizes that "*missiles*" is the subject (`:ARG0 m2/missile`) of "*reach*" in this example. Thus AMR features are able to help predict the correct event type in this case.

AMR can also analyze and learn from different forms of the same word. For example, there are two examples in the ACE corpus involving "*repay*", one using the verb ("*repaying*") and the other one using the noun ("*repayment*"), and both are classified as *Transaction:Transfer-money* event. AMR could learn from the "*repaying*" example about the correct event type and then precisely apply it to the "*repayment*" example.

The gains from adding AMR features show that the features and knowledge encoded in the AMR parse graphs can complement the information incorporated in the dependency parse trees and other traditional features.

[4] Entity mentions and types may get used to introduce more features into the systems.

[5] Global features are the features generated from several event trigger candidates, such as bigrams of trigger types which occur in the same sentence or the same clause, binary feature indicating whether synonyms in the same sentence have the same trigger label, context and dependency paths between two triggers conjuncted with their types, etc.

[6] Because of the limited space, only a subset of event types is listed in Table 3.

Event Type	Baseline			Baseline + AMR		
	P	*R*	F_1	*P*	*R*	F_1
Transaction:Transfer-Ownership	50.0	11.1	18.2	62.5	18.5	28.6
Business:Start-Org	0.0	0.0	0.0	100.0	5.9	11.1
Justice:Trial-Hearing	80.0	80.0	80.0	83.3	100.0	90.9
Justice:Appeal	85.7	100.0	92.3	100.0	100.0	100.0
Conflict:Demonstrate	80.0	57.1	66.7	100.0	57.1	72.8
Justice:Arrest-Jail	75.0	50.0	60.0	83.3	83.3	83.3
Contact:Phone-Write	20.0	12.5	15.4	40.0	25.0	30.8
Personnel:Start-Position	80.0	33.3	47.1	66.7	33.3	44.4
Justice:Release-Parole	50.0	100.0	66.7	33.3	100.0	50.0
Contact:Meet	85.7	87.1	86.4	82.3	82.3	82.3

Table 3: Comparison between the performance (%) of baseline and AMR on a subset of event types.

4.2 Discussion

Applying the AMR features separately, we find that the features extracted from the sibling nodes are the best predictors of correctness, which indicates that the contexts of sibling nodes associated with the AMR tags can provide better evidence for word sense disambiguation of the trigger candidate as needed for event type classification. Features from the parent node and children nodes are also significant contributors.

Performance of the current AMR parser suffers from a lack of training data. For example,

1. *A tank* ***fired*** *on the Palestine Hotel.*
2. *The company* ***fired*** *its president.*

where two "*fired*" are assigned the same PropBank frame (a very coarse notion of word sense), "`fire-01`", rather than distinguishing the different senses here. As measured in the *JAMR* description paper (Flanigan et al., 2014), this parser only achieves 58% in F_1 on the test data using the full pipeline (concept identification and relation identification stages). An AMR parser trained on a larger corpus would help much more on this ED task and other Information Extraction tasks.

5 Related Work

Early research on event detection has primarily focused on local sentence-level representation of trigger candidates in a pipeline architecture (Grishman et al., 2005; Ahn, 2006). Meanwhile, higher level features have been investigated to improve the performance, including: Ji and Grishman (2008); Gupta and Ji (2009); Patwardhan and Riloff (2009); Liao and Grishman (2010; 2011); Hong et al. (2011); McClosky et al. (2011); Huang and Riloff (2012); Li et al. (2012), and Li et al. (2013a). Besides, some recent research has worked on joint models, including methods based on Markov Logic Networks (Riedel et al., 2009; Poon and Vanderwende, 2010; Venugopal et al., 2014), structured perceptrons (Li et al., 2013b), and dual decomposition (Riedel and McCallum (2009; 2011a; 2011b)). However, all of these methods as mentioned above have not exploited the knowledge captured in the AMR.

A growing number of researchers are studying how to incorporate the knowledge encoded in the AMR parse and representations to help solve other NLP problems, such as entity linking (Pan et al., 2015), machine translation (Jones et al., 2015), and summarization (Liu et al., 2015). Especially the appearance of the first published AMR parser (Flanigan et al., 2014) will benefit and spur a lot of new research conducted using AMR.

6 Conclusion and Future Work

Event Detection requires a representation of the relations between the event trigger word and entities in text. We demonstrate that Abstract Meaning Representation can capture deeper contexts of trigger words in this task, and the experimental results show that adding AMR features on top of the traditional features can achieve 67.8% in F-measure with 2.1% absolute improvement over the baseline features. We show that AMR enables ED performance to become comparable to the state-of-the-art approaches.

In this work, we have only applied a subset of AMR representations to the ED task, so we aim to explore more AMR knowledge to be utilized in this task and other Information Extraction tasks, e.g., event argument identification and argument role classification. Furthermore, we are also interested in using AMR knowledge in different machine learning frameworks, such as incorporating the AMR into the SVM tree kernel.

References

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

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of ACL 2013 Workshop on Linguistic Annotation and Interoperability with Discourse*.

Bonnie Dorr, Nizar Habash, and David Traum. 1998. A thematic hierarchy for efficient generation from lexical-conceptual structure. In *Proceedings of the Third Conference of the Association for Machine Translation in the Americas, AMTA-98, in Lecture Notes in Artificial Intelligence*, pages 333–343.

Jeffrey Flanigan, Sam Thomson, Jaime G. Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of ACL*, pages 1426–1436.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. NYU's english ACE 2005 system description. In *ACE 2005 Evaluation Workshop*.

Prashant Gupta and Heng Ji. 2009. Predicting unknown time arguments based on cross-event propagation. In *Proceedings of ACL-IJCNLP*, pages 369–372.

Yu Hong, Jianfeng Zhang, Bin Ma, Jian-Min Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of ACL*, pages 1127–1136.

Ruihong Huang and Ellen Riloff. 2012. Modeling textual cohesion for event extraction. In *Proceedings of AAAI*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL*, pages 254–262.

Bevan Jones, Jacob Andreas, Daniel Bauer, Karl Moritz Hermann, and Kevin Knight. 2015. Semantics-based machine translation with hyperedge replacement grammars. In *Proceedings of COLING*.

Paul Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *Proceedings of LREC*.

Peifeng Li, Guodong Zhou, Qiaoming Zhu, and Libin Hou. 2012. Employing compositional semantics and discourse consistency in chinese event extraction. In *Proceedings of EMNLP*, pages 1006–1016.

Peifeng Li, Qiaoming Zhu, and Guodong Zhou. 2013a. Argument inference from relevant event mentions in chinese argument extraction. In *Proceedings of ACL*, pages 1477–1487.

Qi Li, Heng Ji, and Liang Huang. 2013b. Joint event extraction via structured prediction with global features. In *Proceedings of ACL*, pages 73–82.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of ACL*, pages 789–797.

Shasha Liao and Ralph Grishman. 2011. Acquiring topic features to improve event extraction: in preselected and balanced collections. In *Proceedings of RANLP*, pages 9–16.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of NAACL*.

David McClosky, Mihai Surdeanu, and Chris Manning. 2011. Event extraction as dependency parsing. In *Proceedings of ACL*, pages 1626–1635.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31.

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 2015. Unsupervised entity linking with abstract meaning representation. In *Proceedings of NAACL-HLT*.

Siddharth Patwardhan and Ellen Rilof. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of EMNLP*, pages 151–160.

Hoifung Poon and Lucy Vanderwende. 2010. Joint inference for knowledge extraction from biomedical literature. In *Proceedings of NAACL*, pages 813–821.

Sebastian Riedel and Andrew McCallum. 2011a. Fast and robust joint models for biomedical event extraction. In *Proceedings of EMNLP*, pages 1–12.

Sebastian Riedel and Andrew McCallum. 2011b. Robust biomedical event extraction with dual decomposition and minimal domain adaptation. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, pages 46–50.

Sebastian Riedel, Hong-Woo Chun, Toshihisa Takagi, and Jun'ichi Tsujii. 2009. A markov logic approach to bio-molecular event extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 41–49.

Deepak Venugopal, Chen Chen, Vibhav Gogate, and Vincent Ng. 2014. In *Proceddings of EMNLP*.