Incorporating Event Type Priori and Argument Relationship for Event Extraction

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

Event extraction is a particularly challenging task of information extraction. Most previous work rely on patterns for event type priori, or use a bunch of local and global features for the identification and classification process of triggers and arguments. However, event patterns suffer the low recall issue since the real world events have a large variety of representations. In addition, previous work only considers the features of each argument while ignoring the relation between arguments. In this paper, we propose a **R**egularization-**B**ased model to make full use of the relation between arguments and trained a SVM model to make better use of **E**vent **T**ype priori (RBET method). Experiments show that we achieved a rather good result than the current state-of-art work.

1 Introduction

Event extraction has become a popular research topic in the area of information extraction. ACE 2005 defines the event extraction task[1] as three sub-tasks: identifying the trigger of an event, identifying the arguments of the event and distinguishing their corresponding roles. As an example in Figure 1, there is an "Attack" event triggered by "tear through" with four arguments, each argument has one role.

Previous work can be classified into two folds: (1) pipelined models using event patterns as well as local features (2) the joint models using both local and global features

The first kind of approaches (Grishman et al., 2005; Ji and Grishman, 2008; Liao and Grishman, 2010; Huang and Riloff, 2012) heavily rely on event patterns. Specifically, event patterns are used as event type priori for identifying event trigger and arguments. However, event patterns suffer the low recall issue since the real world events have a large variety of representations. We have carefully studied a pipeline system namely JET[2] which was proposed by Grishman et al. (2005) and followed by Ji and Grishman (2008), Liao and Grishman (2010) and Huang and Riloff (2012). The experimental results show that the recall of trigger identification and classification is relatively low, about 50%$\sim$60%. The main reason is that most of the missed triggers cannot be matched by any pattern. The missing match problem of event patterns will lead errors of the downstream modules (e.g. argument identification and classification). To solve the problem, we trained a classifier to assign the event a type without solely relying on event patterns.

The other kind of approach (Ji and Grishman, 2008; Liao and Grishman, 2010; Li et al., 2013; Lu and Roth, 2012) tried to identify both event trigger and argument in joint ways by incorporating global features (or distributional features). The joint models reduce the cascading errors in the pipeline system. However, both the pipeline systems and joint models identify each candidate argument separately without considering the relation among arguments. The relation here means that how possible that two candidate arguments belong to the same event. Ignoring the relation of arguments mainly lead to the following two issues: (1) Some arguments are not identified, although they have relations with other identified arguments. As the example shown in Figure 1, the underlined arguments are the ground truth, and the arguments circled by dashed line are the identification result of JET. We can see that JET missed "a waiting shed". The entity "a waiting shed" has common dependency governor with "a powerful bomb", so when the latter entity is identified

[1] http://www.itl.nist.gov/iad/mig/tests/ace/2005/

[2] http://cs.nyu.edu/grishman/jet/jet.html

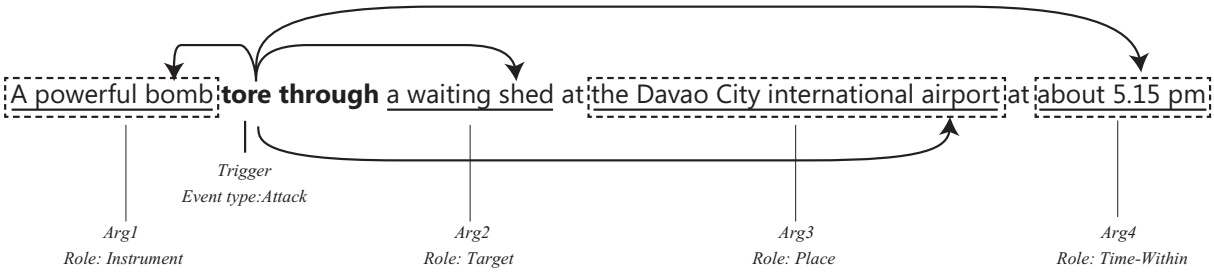


Figure 1: Event example: There is an event trigger by "tear through" with four arguments

as argument, the former should be more likely to be identified. (2) Some arguments are incorrectly identified even they are irrelevant to the event. We have carefully studied the events in which the arguments are wrongly identified. Among the 641 arguments identified by JET on ACE 2005 testing corpus, 106 arguments were incorrectly identified. The connections between the correctly identified arguments and the incorrectly identified arguments are very weak. This means that they cannot be identified simultaneously if argument relations can be leveraged.

In summary, we first train an event type classifier to predict the event type to address the missing match problem of event patterns. Second, we use a regularization method to model the relation arguments to improve the argument identification. Our approach is named RBET.

The contribution of this paper is as follows:

- We implemented an event type classifier for incorporating event type priori to event extraction. If no patterns can be matched, the classifier can predict the event type based on the local features in the sentence, which helps to identify and classify the arguments.
- We proposed a regularization method in order to make full use of the relation between candidate arguments. The regularization method improves the performance of argument identification.

2 Related Work

Event extraction is very important in the knowledge mining field. An event consists of a trigger and several arguments. To extract an event is to identify the trigger and arguments from raw text, then label the arguments with correct role, and finally classify the event into the correct event type. Previous works can be classified to two kinds of methods: (1) pipelined model using event patterns as well as local features (2) methods using local and global features.

Many classical work focus on local text, and use pattern based method for event type priori (Kim and Moldovan, 1993; Riloff and others, 1993; Soderland et al., 1995; Huffman, 1996; Freitag, 1998b; Ciravegna and others, 2001; Califf and Mooney, 2003; Riloff, 1996; Riloff et al., 1999; Yangarber et al., 2000; Sudo et al., 2003; Stevenson and Greenwood, 2005; Grishman et al., 2005; Ji and Grishman, 2008; Liao and Grishman, 2010; Huang and Riloff, 2012), others use local feature based classification method (Freitag, 1998a; Chieu and Ng, 2002; Finn and Kushmerick, 2004; Li et al., 2005; Yu et al., 2005). Moreover, a variety of techniques have been explored for weakly supervised training (pattern-based and rule-based) of event extraction systems (Riloff, 1996; Riloff et al., 1999; Yangarber et al., 2000; Sudo et al., 2003; Stevenson and Greenwood, 2005; Patwardhan and Riloff, 2007; Chambers and Jurafsky, 2011). In some of these systems, human should help to delete some nonsense patterns or rules. (Shinyama and Sekine, 2006; Sekine, 2006) are unsupervised methods to extract patterns from open domain texts. However, pattern is not always enough although some method (Huang and Riloff, 2012; Liu and Strzalkowski, 2012) use bootstrapping to get more patterns.

Some other methods (Gu and Cercone, 2006; Patwardhan and Riloff, 2009) considered broader context when deciding the role fillers. Other systems take the whole discourse feature into consideration, such as (Maslennikov and Chua, 2007; Liao and Grishman, 2010; Hong et al., 2011; Huang and Riloff, 2011). Ji and Grishman (2008) even considers the topic-related documents, so they proposed the cross-document method. Liao and Grishman (2010; Hong et al. (2011) use a series of global features (for example, the occurence of one event type lead to the occurence of another)

to improve the role assignment and event classification performance.

Joint models (Li et al., 2013; Lu and Roth, 2012) is also an important contribution. Li et al. (2013) makes full use of the local feature and global feature to get a better result. The semi-CRF based method (Lu and Roth, 2012) trains separate models for each event type, which requires a lot of training data. However, all of these above methods considers arguments separately while ignoring the relation between arguments.

In addition, Ritter et al. (2012), Zhou et al. (2014) and Zhou et al. (2015) used unsupervised method to extract events from Twitter.

In summary, most of the above works are strongly depended on patterns which will bring severe loss of recall. In addition, the arguments are considered independently while the relation between arguments is also important for argument identification. We implement an event type classifier and designed a regularization method to solve the two problems.

3 ACE Event Extraction Task

Automatic Content Extraction (ACE) is an event extraction task. It annotates 8 types and 33 subtypes of events. ACE defines the following terminologies:

- Entity: an object or a set of objects in one of the semantic categories of interest
- Entity mention: a reference to an entity, usually a noun phrase (NP)
- Event trigger: the main word which most clearly expresses an event occurrence
- Event arguments: the entity mentions that are involved in an event
- Argument roles: the relation of arguments to the event where they participate, there are 35 roles in total
- Event mention: a phrase or sentence within which an event is described, including trigger and arguments

Given an English document, an event extraction system should identify event triggers with their subtypes and their arguments from each sentence. An example is shown in Figure 1, there is an "Attack" event triggered by "tear through" with four arguments, each argument has a role type such as "Instrument", "Target", etc.

For the evaluation metric, we follow the previous works (Ji and Grishman, 2008; Liao and Grishman, 2010; Li et al., 2013). We use the following criteria to determine the correctness of the predicted event mentions.

- A trigger is considered to be correct if and only if its event type and offsets can match the reference trigger;
- An argument is correctly identified if and only if its event type and offsets can match any of the reference arguments;
- An argument is correctly identified and classified if and only if its event type, offsets and role can match any of the reference arguments.

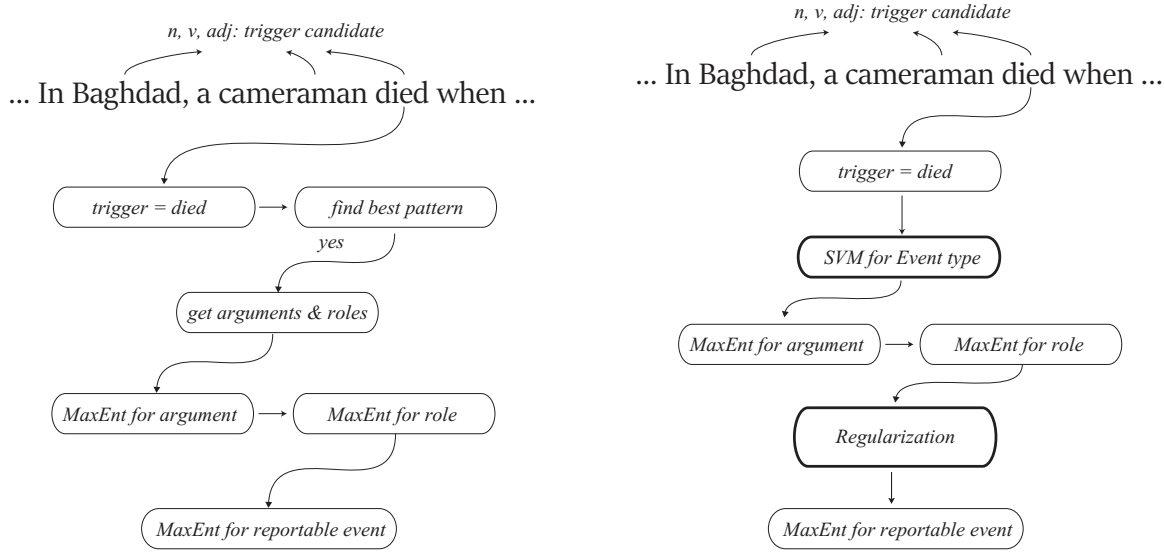
4 Baseline: JET Extractor for Events

Many previous works take JET as their baseline system, including Ji and Grishman (2008), Liao and Grishman (2010), Li et al. (2013). JET extracts events independently for each sentence. This system combines pattern matching with statistical model together. For each event mention in the training corpus of ACE, the patterns are constructed based on the sequences of constituent heads separating the trigger and arguments. After that, three Maximum Entropy classifier are trained using the local features as well as the pattern match features:

- Argument Classifier: to distinguish arguments from non-arguments
- Role Classifier: to classify arguments by argument role
- Reportable-Event Classifier: to determine whether there is a reportable event mention according to the trigger, event type, and a set of arguments

The features the three classifier used are listed in Table 1.

Figure 2(a) shows the whole test procedure. In the test procedure, each sentence is scanned for nouns, verbs and adjectives as trigger candidates. When a trigger candidate is found, the system tries to match the context of the trigger against the set of patterns associated with that trigger. If this pattern matching process is successful, the best pattern will assign some of the mentions in the sentence as the arguments of a potential event mention. For the remaining mentions in the sentence, the argument classifier is applied; for any argument pass-



(a) The flow chart of JET

(b) The flow chart of our approach

Figure 2: The left is the flow chart of JET, the right is the flow chart of our approach, the thick line block is our contribution

	Origin	Added
Arg	Trigger, EvType, MentionType, argument, EvTypeAndArg, EvTypeAndMentionType, prevToken, prevTokenAndType, chunkPathAndType synPathEvType	Can Match best pattern
Role	Trigger, EvType, MentionType, argument, EvTypeAndArg, EvTypeAndMentionType, prevToken, prevTokenAndType, chunkPathAndType synPathEvType	Can Match best pattern
Event	Trigger, Pattern match score	EvType TriggerPOS

Table 1: The Maximum Entropy Classifier features

ing that classifier, a role is assigned to it using the role classifier. Finally, once all arguments have been assigned, the reportable-event classifier is applied to decide whether this event mention should be reported.

5 Motivations

5.1 Error Analysis

We have carefully studied the JET system. After carefully analysis the errors of JET, we found that they can be attributed to two types of errors. First, the pattern set is not always enough for so large variety of events, so that the event type priori cannot be fully utilized, which could lead to severe recall loss. Second, the arguments are considered independently, while the relation between candidate arguments is also very important to the argument identification.

		Trigger Classification	Argument Classification	Role Classification
JET	P	67.56	46.45	41.02
	R	53.54	37.15	32.81
	F	59.74	41.29	36.46

Table 2: Overall performance of JET on blind test data

5.1.1 Trigger Error

The performance of JET is shown in Table 2. According to the table, we found that the recall of "Trigger Classification" is only 53.54%, which means about 46.46% event mentions are lost. We randomly sample 40 documents in ACE corpus and use the origin JET system to extract events for statistical analysis. According to statistics, of all the lost event mentions, about 96.3% are due to the missing of corresponding event pattern, while the last 3.7% have a matched pattern but the event type is wrong.

The error in the trigger identification and classification can affect the argument identification and classification severely. In the statistical result, there are in total 230 argument identification error. Among them, 124 are due to the trigger error, which took about 53.9%.

For the 96.3% triggers which do not have corresponding patterns, their event type cannot be correctly assigned. Then, the arguments cannot be correctly identified and classified. Hence, for those triggers without any matched patterns, the

event type should be correctly predicted, and then this event type may help to identify and classify arguments together with other features. This method may supply the patters in some extent.

5.1.2 Argument Error

Apart from the 124 trigger-caused argument identification error, the remaining 106 errors can be split into two types of error.

- Error 1: An error argument has been identified
- Error 2: The right argument has not been identified

Intuitively, we can observe that if two entities belong to the arguments of the same event, there are always some obvious relations between them. For example, "Police have arrested four people in connection with the killings", the argument "police" and "four people" share the same father in the dependency parse tree. Certainly, the actual situation is much more complicated than this, but it is sure that the arguments of one event have relation with each other.

Corresponding to the two kinds of errors, we should capture two kinds of argument relationship. (1) Two arguments should occur in the same event; (2) Two argument cannot occur in the same event. For the first relationship, if one argument is identified, then the other is more likely to be identified. For the second relationship, if one argument is identified, then the other cannot be identified. Intuitively, with these two kinds of relationship, the argument identify performance will improve.

6 Approach Overview

Based on the above analysis, we decide to make two improvements: (1) make better use of event type priori: if there is no corresponding pattern, instead of cast it away immediately, we use a classifier to assign the current trigger an event type and assign some of the entity mentions as its arguments, (2) use a regularization method to make full use of the relation of arguments.

The thick line blocks in Figure 2(b) depicts our improvements. Whether or not any pattern can be matched, we identify and classify the arguments directly by the maximum entropy classifier instead of by patterns first. The best matched pattern and the event type we got from the classifier can be taken as a feature in the following maximum entropy classifiers. After the outputs of argument and role classifier are calculated, we make use of the argument relationship to regularize for a better result. In addition, we add some new features to the origin maximum entropy classifier as is shown in Table 1 in order to make the origin classifier more suitable for our improvement.

6.1 Incorporating Event Type Priori

Event type is very important priori information for argument identification and role classification, not only the event type is an important feature, the corresponding event schema can also help the argument identification and role classification a lot. We would like to assign an event type to the given trigger based on the sentence it is located.

We use WORD2VEC[3] for an embedding representation of common words, the word vectors are trained on the default "text8" training text data. Each word vector is 200-dim.

The feature of the event type classifier is a 400-dim vector, which is concatenated by two word vectors. The first is the trigger's word vector. The second is denoted as $Aver(NP)$ and can be calculated by Eq 1, which represents the information of all the candidate arguments.

$$Aver(NP) = \frac{1}{n}\sum_{i} Vec(Head(NP_i)) \quad (1)$$

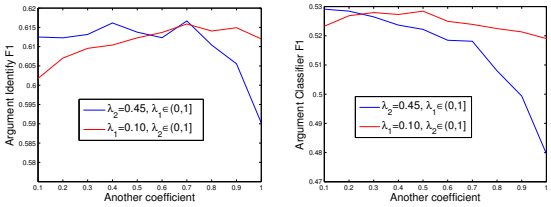
Here, the $Vec(\cdot)$ means a word's vector representation. $Head(\cdot)$ means the head word of an NP (for example, "a little girl" has a head word of "girl").

We trained a SVM classifier using the training documents, the classifying precision on the test documents achieved 80.6%. We have carefully checked the wrong cases. We found that the meanings of some event types are rather similar like "Transfer-Ownership" and "Transfer-Money", which caused many errors. Hence, we did not continue to improve this classifier.

6.2 Capture the Relationship Between Arguments

In this section, we will capture two kinds of relation between arguments described in Section 5.1.2. For a trigger, if there are n candidate arguments, we set a $n \times n$ matrix C to represent the relationship between arguments. If $C_{i,j} = 1$, then argument i and argument j should belong to

[3] http://code.google.com/p/word2vec/



(a) Arg Identify (b) Arg Classify

Figure 3: The trend graph when fix one coefficient and change another

the same event. If $C_{i,j} = -1$, then argument i and argument j cannot belong to the same event.

We use a n-dim vector X to represent the identification result of arguments. Each entry of X is 0 or 1, 0 represents "noArg", 1 represents "arg". The evaluation of X should be calculated by Eq 2. The larger $E(X)$ is, the better X will be.

$$E(X) = \lambda_1 X^T C X + \lambda_2 P_{sum}^{arg} + (1 - \lambda_1 - \lambda_2) P_{sum}^{role} \quad (2)$$

Here, $X^T C X$ means to add up all the relationship value if the two arguments are identified. Hence, the more the identified arguments are related, the larger the value $X^T C X$ is. P_{sum}^{arg} is the sum of all chosen arguments probability. The probability here is the output of the arguments' maximum entropy classifier. P_{sum}^{role} is the sum of all the classified roles' probability. The probability here is the output of the roles' maximum entropy classifier. We tuned the coefficients λ_1 and λ_2 on the development set, and finally we set $\lambda_1 = 0.10$ and $\lambda_2 = 0.45$. Figure 3 shows the variation of argument identification's F1 measure and argument classification's F1 measure when fix one parameter and change another. Note that the third coefficient $1 - \lambda_1 - \lambda_2$ must be positive, which is the reason why the curve decreases sharply when λ_2 is fixed and $\lambda_1 > 0.65$.

Eq 2 means that, while we should identify and classify the candidate arguments with larger probability, the argument relationship should also as much as possible. The arguments should also follow the following constraints. These constraints together with Eq 2 can make the argument identification and classification help each other for a better result.

- Each entity can only take one role
- Each role can belong to one or more entities
- The role assignment must follow the event schema of the corresponding type

We use Beam Search method to search for the optimal assignment X.

6.2.1 Training the Argument Relationship Matrix

The argument relationship matrix is very important in the regularization process. We trained a maximum entropy classifier to predict the connection between two entities. The entity pairs in the ground truth events are used for our training data. After a large amount of trials, we choose the following features:

- TRIGGER: the trigger of the event
- ENTITY DISTANCE: the distance between the two candidate arguments in the sentence
- Whether the two candidate arguments occur on the same side of the trigger
- PARENT DEPENDENCY DISTANCE: the distance between the two candidate arguments' parents in the dependency parse tree
- PARENT POS: if the two candidate arguments share the same parent, take the common parent's POS tag as a feature
- Whether the two candidate arguments occur on the same side of the common parent if the two candidate arguments share the same parent

For an entity pair, if both of the entities belong to the same event's arguments, we take it as positive example. For each positive example, we randomly exchange one of the entities with an irrelevant entity (an irrelevant entity is in the same sentence with the event, but it is not the event's argument) to get a negative example. In the testing procedure, we predict the relationship between entity i and entity j using the maximum entropy classifier. Since in many cases, the relations may be very obscure and difficult to be figure out. So we only capture a portion of them. We set two thresholds, if the output of the maximum entropy classifier is larger than 0.8, we set $C_{i,j} = 1$, if the output is lower than 0.2, we set $C_{i,j} = -1$.

7 Experiments

7.1 Data

We utilize ACE 2005 data sets as our testbed. As is consistent with previous work, we randomly select 10 newswire texts from ACE 2005 training corpora as our development set, and then conduct blind test on a separate set of 40 ACE 2005

Abbreviation	Illustration
JET	The within-one-sentence baseline of Grishman et al. (2005)
CD	Cross-Document (Ji and Grishman, 2008)
CEV	Cross-Event (Liao and Grishman, 2010)
CEN	Cross-Entity (Hong et al., 2011)
Joint	Joint model in (Li et al., 2013)
SP	Unsupervised Structured Preference method in (Lu and Roth, 2012)
semi-CRF	Supervised CRF in (Lu and Roth, 2012)
RBET	Our approach with both event type classifier and regularization
RBET-Regu	Our approach without regularization
RBET-ET	Our approach without event type classifier

Table 3: Baseline illustration

Method	Trigger F1	Arg id F1	Arg id+cl F1
JET	59.7	42.5	36.6
CD	67.3	46.2	42.6
Joint	65.6	-	41.8
RBET	66.0	**55.4**	**43.8**
RBET-Regu	66.0	51.8	42.7
RBET-ET	64.8	54.6	43.0

Table 5: Overall Performance with predicted entities, timex, and values

newswire texts. The rest of ACE training corpus is used as the training data.

7.2 Baseline System

The abbreviation and simple illustration of the baselines as well as our approach are listed in Table 3. The JET system is the within-one-sentence baseline proposed by (Grishman et al., 2005). The CD, CEV and CEN are all extension systems of the within-one-sentence baseline. The joint model (Li et al., 2013) is the currently state-of-art method. Among these methods, CEV and CEN make use of the gold-standard entities, timex, and values annotated in the corpus as the argument candidates. CD uses the JET system to extract the candidate arguments. Li et al. (2013) reports the performance with both gold-standard argument candidates and predicted argument candidates. Therefore, we compared our result with methods based on gold argument candidates in Table 4 and methods based on predicted argument candidates in Table 5. The structured preference method and semi-CRF method are proposed by (Lu and Roth, 2012). Since the semi-CRF requires large amount of training data, the author chose to perform their evaluations on the top 4 events ("Attack", "Meet", "Die" and "Transport") which have the most instances. We compared our result with them in Table 6.

7.3 Overall Performance

We conducted experiments to answer the following questions. (1) Can the event type classifier lead to a higher recall in trigger classification, argument identification and classification while retaining the precision value? (2) Can the regularization step improve the performance of argument identification and classification?

Table 4 shows the overall performance on the blind test set. We compared our result with the JET baseline as well as the CEV, CEN and joint methods. When added the event type classifier, in the line named "RBET-Regu", we gain a significant increase in the three measures over the JET baseline in recall. Although our trigger's precision is lower than JET, it gains 6.1% improvement on trigger's F1 measure, 19.5% improvement on argument identification's F1 measure and 11.2% improvement on argument classification's F1 measure. In the predicted argument candidate situation in Table 5, our approach "RBET-Regu" again significantly outperforms the JET baseline. Remarkably, our result is comparable with Joint model although we only use local features. The line with name "RBET-ET" in Table 4 and Table 5 represents the performance when we only use regularization method. In Table 4, Compared to the four baseline systems, the argument identification's F1 measure of "RBET-ET" is significantly higher. In Table 5, the "RBET-ET" again gains a higher F1 measure than the JET, CD, joint model baseline and "RBET-Regu".

The complete approach is denoted as "RBET" in Table 4 and Table 5. Remarkably, our approach performances comparable in trigger classification with the state-of art methods: CD, CEV, CEN, Joint model, and significantly higher than them in argument identification as well as classification although we did not use the cross-document, cross-event information or any global feature. This may prove that our assume that the relationship between argument candidates can help to improve the argument identification performance is right. The event type classifier also contributes a lot in trigger identification & classification. We have done the Wilcoxon Signed Rank Test on trigger classification, argument identification and argument classification, all the three have $p < 0.01$.

The comparison between our approach and Lu and Roth (2012)'s methods is listed in Table 6. The numbers in the table is the micro average F1

Method	Trigger Classification			Argument Identification			Argument Role		
	P	R	F1	P	R	F1	P	R	F1
JET	67.6	53.5	59.7	46.5	37.2	41.3	41.0	32.8	36.5
CEV	68.7	68.9	68.8	50.9	49.7	50.3	45.1	44.1	44.6
CEN	72.9	64.3	68.3	53.4	52.9	53.1	51.6	45.5	48.3
Joint	73.7	62.3	67.5	69.8	47.9	56.8	64.7	44.4	52.7
RBET	67.2	64.7	65.9	63.2	59.4	**61.2**	54.1	53.5	**53.8**
RBET-Regu	65.7	65.9	65.8	60.6	61.1	60.8	47.2	48.3	47.7
RBET-ET	67.2	61.7	64.3	63.6	57.6	60.5	51.6	47.4	49.4

Table 4: Overall Performance with gold-standard entities, timex, and values

Event	SP	semi-CRF	Our Approach		
			RBET-Regu	RBET-ET	RBET
Attack	42.02	63.11	66.84	69.55	70.36
Meet	63.55	76.64	70.89	71.25	73.04
Die	55.38	67.65	65.56	66.63	68.50
Transport	57.29	64.19	63.72	65.44	67.67

Table 6: Performance compared to (Lu and Roth, 2012) with gold-standard entities, timex, and values

measure of argument classification. We did not train separate models for each event type so that the performance of "Meet" event is lower than semi-CRF but we still outperforms the semi-CRF method in "Attack", "Die" and "Transport" events.

However, our approach is just a pipeline approach which suffers from error propagation and the argument performance may not affect the trigger too much. We can see from Table 4, although we use the gold argument candidates, the trigger performance is still lower than CEV, CEN and joint model. Another limitation is that our regularization method did not improve the argument classification too much since it only uses constraints to affect roles. Future work may be done to solve these two limitations.

8 Conclusion

In this paper, we proposed two improvements based on the event extraction baseline JET. We found that JET depends too much on event patterns for event type priori and JET considers each candidate argument separately. However, patterns cannot cover all events and the relationship between candidate arguments may help when identifying arguments. For a trigger, if no pattern can be matched, the event type cannot be assigned and the arguments cannot be correctly identified and classified. Therefore, we developed an event type classifier to assign the event a type, then identify and classify the arguments using the type information as well as other features.

On the other hand, we trained a maximum entropy classifier to predict the relationship between candidate arguments. Then we proposed a regularization method to make full use of the argument relationship. The experiment result shows that the regularization method brings a significant result in the argument identification over previous works.

In summary, by using the event type classifier and the regularization method, we have achieved a good performance in which the trigger classification is comparable to the state-of-the-art methods, and the argument identification & classification performance is significantly better than state-of-the-art methods. However, we only use the sentence-level features and our method is a pipelined approach. Also, the argument classification seems not to be affected too much by the regularization method. Future work may be done to integrate our method into a joint approach, use some global feature and try to improve the argument classification by the regularization, which may make our performance better.

References

Mary Elaine Califf and Raymond J Mooney. 2003. Bottom-up relational learning of pattern matching rules for information extraction. *The Journal of Machine Learning Research*, 4:177–210.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 976–986. Association for Computational Linguistics.

Hai Leong Chieu and Hwee Tou Ng. 2002. A maximum entropy approach to information extraction from semi-structured and free text. *AAAI/IAAI*, 2002:786–791.

Fabio Ciravegna et al. 2001. Adaptive information extraction from text by rule induction and generalisation. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 1251–1256. LAWRENCE ERLBAUM ASSOCIATES LTD.

Aidan Finn and Nicholas Kushmerick. 2004. *Multi-level boundary classification for information extraction*. Springer.

Dayne Freitag. 1998a. Multistrategy learning for information extraction. In *ICML*, pages 161–169.

Dayne Freitag. 1998b. Toward general-purpose learning for information extraction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 404–408. Association for Computational Linguistics.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nyus english ace 2005 system description. *ACE*, 5.

Zhenmei Gu and Nick Cercone. 2006. Segment-based hidden markov models for information extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 481–488. Association for Computational Linguistics.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1127–1136. Association for Computational Linguistics.

Ruihong Huang and Ellen Riloff. 2011. Peeling back the layers: detecting event role fillers in secondary contexts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1137–1147. Association for Computational Linguistics.

Ruihong Huang and Ellen Riloff. 2012. Bootstrapped training of event extraction classifiers. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 286–295. Association for Computational Linguistics.

Scott B Huffman. 1996. Learning information extraction patterns from examples. In *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, pages 246–260. Springer.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of the 46st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 254–262.

Jun-Tae Kim and Dan I Moldovan. 1993. Acquisition of semantic patterns for information extraction from corpora. In *Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on*, pages 171–176. IEEE.

Yaoyong Li, Kalina Bontcheva, and Hamish Cunningham. 2005. Using uneven margins svm and perceptron for information extraction. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 72–79. Association for Computational Linguistics.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria, August. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797. Association for Computational Linguistics.

Ting Liu and Tomek Strzalkowski. 2012. Bootstrapping events and relations from text. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 296–305. Association for Computational Linguistics.

Wei Lu and Dan Roth. 2012. Automatic event extraction with structured preference modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 835–844. Association for Computational Linguistics.

Mstislav Maslennikov and Tat-Seng Chua. 2007. A multi-resolution framework for information

extraction from free text. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 592. Citeseer.

Siddharth Patwardhan and Ellen Riloff. 2007. Effective information extraction with semantic affinity patterns and relevant regions. In *EMNLP-CoNLL*, volume 7, pages 717–727. Citeseer.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 151–160. Association for Computational Linguistics.

Ellen Riloff et al. 1993. Automatically constructing a dictionary for information extraction tasks. In *AAAI*, pages 811–816.

Ellen Riloff, Rosie Jones, et al. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*, pages 1044–1049.

Alan Ritter, Oren Etzioni, Sam Clark, et al. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM.

Satoshi Sekine. 2006. On-demand information extraction. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 731–738. Association for Computational Linguistics.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 304–311. Association for Computational Linguistics.

Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy Lehnert. 1995. Crystal: Inducing a conceptual dictionary. *arXiv preprint cmp-lg/9505020*.

Mark Stevenson and Mark A Greenwood. 2005. A semantic approach to ie pattern induction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 379–386. Association for Computational Linguistics.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 224–231. Association for Computational Linguistics.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 940–946. Association for Computational Linguistics.

Kun Yu, Gang Guan, and Ming Zhou. 2005. Resume information extraction with cascaded hybrid model. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 499–506. Association for Computational Linguistics.

Deyu Zhou, Liangyu Chen, and Yulan He. 2014. A simple bayesian modelling approach to event extraction from twitter. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 700–705.

Deyu Zhou, Liangyu Chen, and Yulan He. 2015. An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Evaluation Algorithms for Event Nugget Detection : A Pilot Study

Zhengzhong Liu, Teruko Mitamura, Eduard Hovy
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
liu@cs.cmu.edu, teruko@cs.cmu.edu, hovy@cmu.edu

Abstract

Event Mention detection is the first step in textual event understanding. Proper evaluation is important for modern natural language processing tasks. In this paper, we present our evaluation algorithm and results during the Event Mention Evaluation pilot study. We analyze the problems of evaluating multiple event mention attributes and discontinuous event mention spans. In addition, we identify a few limitations in the evaluation algorithm used for the pilot task and propose some potential improvements.

1 Introduction

Textual event understanding has attracted a lot of attention in the community. Recent work has covered several areas about events, such as event mention detection(Li et al., 2013; Li et al., 2014) , event coreference (Bejan et al., 2005; Chen and Ji, 2009; Lee et al., 2012; Chen and Ng, 2013; Liu et al., 2013), and script understanding (Chambers and Jurafsky, 2008; Chambers and Jurafsky, 2009). Event Mention detection is the fundamental preprocessing step for these tasks. However, downstream event researches often make minimal effort for event mention detection. For example, in event coreference work, Lee et al. (2012) do not make clear distinction between event and entity mentions. Bejan et al. (2005) and Liu et al. (2013) use oracle event mentions from human annotations. Building robust event mention detection system can help promote research in these areas and enable researchers to produce end-to-end systems. In this paper, we discuss our recent effort in providing a proper evaluation metric for event mention detection.

1.1 The Event Nugget Detection Task

As defined in Mitamura (2014), event nugget detection involves identifying semantic meaningful units (**mention span detection**) that refer to an event[1]. The task also requires a system to identify other attributes (**attribute detection**). In this pilot study, the attributes are *event type* and *realis status*.

(1) President Obama will `nominate` [realis: Other type: Personnel.Nominate] John Kerry for Secretary of State.

(2) He `carried out` the `assassination` [realis: Actual type: Life.Die] .

Example 1 shows one annotated event nugget `nominate`, which has the realis type "other" and event type "Personnel.Nominate". Example 2 annotates one event nugget with discontinuous event span `carried out assassination`. The evaluation corpus is annotated with event nuggets that fall into 8 types of event[2]. Please refer to Mitamura (2014) for detailed definitions of the attributes.

1.2 Past Evaluation Methods

The Automatic Content Extraction 2005 evaluation task involves event extraction. The Event Detection and Recognition (VDR) task in the Automatic Content Extraction 2005 evaluation (NIST, 2005) evaluate the accuracy of event arguments and multiple other event attributes. However, event mention recognition is not directly evaluated (§3.2).

[1]This is similar to Event Trigger in ACE 2005, which is adopted in other work (Li et al., 2013; Li et al., 2014)

[2]These are *Life, Movement, Business, Conflict, Contact, Personnel, Transaction, Justice*

Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT 2015, pages 53–57,
Denver, Colorado, June 4, 2015.

Li et al. (2013; 2014) evaluate event trigger detection using a mention-wise F-1 score. An event trigger is considered correct only when the span and event type are matched exactly. Errors from different sources are not separately presented.

In addition, most previous evaluations on event mention evaluation do not give partial credits to partial matches. Partial scoring is more important in the current setting because of the mention span detection task is difficult with discontinuous event nuggets.

2 The Evaluation Algorithm in Pilot Study

In this section, we describe our mention detection algorithms[3]. We will use the terms Event Nugget and Event Mention interchangeably.

2.1 Prerequisites

The main prerequisite for the evaluation is tokenization. In our pilot study, we provide a standard tokenization for all participants. System responses represent each event mention in terms of predefined token ids[4]. Discontinuous mentions can be easily represented using tokens.

2.2 Partial Span Scoring

The proposed evaluation produces a span similarity score for a pair of mentions (system and gold standard) between 0 and 1. Given a pair of mentions (G, S), we represent the span of each mention by a set of token ids (T_G , T_S). The span similarity score is defined as the Dice coefficient between the two sets (which is the same as the F-1 score).

$$Dice(T_G, T_S) = \frac{2|T_G T_S|}{|T_G| + |T_S|}$$
$$= \frac{2}{|T_G|/|T_G T_S| + |T_S|/|T_G T_S|}$$
$$= F1(T_G, T_S) = \frac{2}{1/P + 1/R}$$

2.3 Mention Mapping

To evaluate mention attributes, the evaluation algorithm needs to decide which system mention corresponds to a gold standard mention. We refer to this step as mention mapping. The input of our mention-mapping algorithm is the pairwise scores between all gold standard vs. system mention pair. We use the token-based Dice score (§2.2). Algorithm 1 shows our mapping algorithm to compute the mapping in one document.

Algorithm 1 Compute a mapping between system and gold standard mentions

Input: A list L of scores $Dice(T_G, T_S)$ for all pair of G, S in the document

1: $M \leftarrow \varnothing; U \leftarrow \varnothing$
2: **while** $L \neq \varnothing$ **do**
3: $\quad G_m, S_n \leftarrow \arg\max_{(G,S)\in L} Dice(T_G, T_S)$
4: $\quad$ **if** $S_n \notin U$ **and** $Dice(T_{G_m}, T_{S_n}) > 0$ **then**
5: $\quad\quad M_{G_m} \leftarrow M_{G_m} \cup (S_n, Dice(T_{G_m}, T_{S_n}))$
6: $\quad\quad U \leftarrow U \cup \{S_n\}$

Output: The mapping M

Algorithm 1 iteratively searches for the highest Dice score in all remaining mention pairs. Line 4 ensures that each system mention can only be mapped to one gold standard mention to avoid multiple counting. One gold standard mention is allowed to be mapped to multiple system mentions, which will be used in calculating attribute accuracy scores.

2.4 Overall Span Scoring

In the pilot study, we first evaluate the system's performance on span detection[5]. We use F-1 score (referred as mention level F-1 score to distinguish with the token level F-1 score in §2.2) for this task.

The definition of True Positive (TP) and False Positive (FP) for mention-level F-1 are slightly adjusted to reflect partial matching. TP values are accumulated according to Algorithm 2.

Precision, Recall, F-1 are calculated as followed:

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

N_G is the number of gold standard mentions.

In the study, we use $TP + FP$ as the denominator for Precision. We later identify a problem of this formulat. When FP is 0, even if the span range is

[3]Code base: `github.com/hunterhector/EvmEval`

[4]Some other KBP evaluations use character span evaluation, which will favor long words than short words. We argue that the difficulties in tokenizing a long word and a short word in English should be virtually the same; hence scoring these two cases differently is not fair.

[5]For simplicity, we describe our algorithm on a single document, the scorer will produce aggregate results for each metric with standard Micro and Macro average methods.

Algorithm 2 Compute TP and FP

Input: The set of gold standard $\mathcal{G}$; The mapping M indexed by G; Number of system mentions N_S

1: $TP \leftarrow 0$; $FP \leftarrow 0$
2: **for** $\forall G \in \mathcal{G}$ **do**
3: **if** $|M_G| = 0$ **then**
4: $FP \leftarrow FP + 1$
5: **else**
6: $S_T \leftarrow \arg\max_{Dice}(S, Dice) \in M_G$
7: $TP \leftarrow TP + Dice(G, S_T)$

Output: TP

not exactly correct, the system can still get perfect precision (though imperfect recall), which is counter-intuitive. If we calculate FP with $N_S - TP$, the precision, recall calculation will naturally resolve to:

$$P = \frac{TP}{N_S}; R = \frac{TP}{N_G}$$

The new formula is also aesthetically symmetric on precision and recall. We present the influence of this fix in §4.1.

2.5 Attribute Scoring

For each attribute and gold standard mention, we calculate the accuracy according to algorithm 3. This algorithm will give a system full credit even when the span matching is not perfect. In addition, when one system incorrectly splits one gold standard mention into two, we still give it credit as long as attributes are all predicted correctly.

Algorithm 3 Compute Attribute Accuracy for one Gold Standard Mention

Input: The gold standard mention G; The mapping M indexed by G; The set A indexing target attributes for all mentions;

1: $Accuracy \leftarrow 0$
2: **for** $S, Dice(T_S, T_G) \in M_G$ **do**
3: **if** $A_S = A_G$ **then**
4: $Accuracy \leftarrow Accuracy + 1/|M_G|$

Output: $Accuracy$

Gold He `carried out` the `assassination` [type: Life.Die] .

System 1 He `carried`[type: Life.Die] out the `assassination` [type: Life.Die] .

System 2 He `carried`[type: Business.MERGE] out the `assassination` [type: Life.Die].

In the above examples, there is one gold standard mention while both systems report two event mentions, and they both omit the word "out". According algorithm 3, **System 1** gets full credit while **System 2** gets 0.5. The algorithm is designed this way to prevent a system being penalized again for its span error. However, this make it difficult to find a natural way to combine span scores with attribute scores.

2.6 Combining multiple scores

Algorithm 2 and 3 are limited in that there is no one simple score for final system ranking. Furthermore, the span score only reflects the system's ability to distinguish the 8 types of event mentions from everything else, which is not a useful metric by its own.

A naive way to combine the scores is to multiply these individual scores. However, theoretically, the errors in attribute scoring and the span scoring are not independent, thus it is inappropriate to perform a simple multiplication. We propose a natural adjustment by directly augmenting attribute evaluation into F1 score calculation (Algorithm 4). Line 3 in the algorithm finds a system mention with the highest mapping score that also fits all the attributes of interest as true positive. We can choose the set $\mathcal{A}$ to contain the desired attributes we would like to evaluate on. In our implementation, we iterate all possible attribute combinations and produce all the scores (§4.2).

Algorithm 4 Compute True Postive with Attributes

Input: The set of gold standard mentions $\mathcal{G}$; The mapping M indexed by gold standard mentions; Number of system mentions N_S; The set $\mathcal{A}$ indexing the attributes that will be evaluated for all mentions

1: $TP \leftarrow 0$
2: **for** $G \in \mathcal{G}$ **do**
3: $S_{max} \leftarrow \arg\max_{Dice}(S, Dice) \in M_G$ Subject to $\mathcal{A}_{S_{max}} = \mathcal{A}_G$
4: $TP \leftarrow TP + Dice(S_{max}, G)$

Output: TP

3 Comparison with Previous Methods

3.1 Comparison with MUC

The Message Understanding Conference provides a scoring algorithm for the information extraction task (Chinchor, 1992). Though there is no event mention evaluation, some algorithm design can still be compared with our methods.

The MUC scorer first calculates an alignment between gold standard mention and system, and then counts the number of exact matches COR, the number of partial matches PAR, the number of gold standard keys POS, the number of system responses ACT. The precision and recall are calculated as[6] :

$$P = \frac{COR + 0.5PAR}{POS}; R = \frac{COR + 0.5PAR}{ACT}$$

The MUC scorer then takes the highest F-Score from all possible alignments.

Our method makes several different decisions. First, we use a simple greedy method for choosing an alignment based on span matching instead of trying to find the best alignment.

Second, we give a partial score between 0 to 1 using the Dice Coefficient, while MUC uses a universal partial credit of 0.5. A variable partial score can reflect more subtle differences between systems.

3.2 Comparison with ACE

The Automatic Content Extraction 2005 task included an event related evaluation (NIST, 2005). The Event Mention Detection (VMD) task described in the evaluation guideline defines the event mention as a sentence or phrase. The ACE event task evaluates the systems on the attributes and arguments of a whole event (which may contains multiple event mentions). Such evaluation also requires a system to resolve event coreference. Thus, there is no direct evaluation for event nuggets in ACE 2005.

4 Experiments

We conduct evaluation on the 15 pilot study submissions using the LDC2015E3 dataset, which contains 200 documents with 6921 annotated event mentions. The results we show in this section are all micro average across these mentions.

[6]We simplified the discussion by assuming there is no optional gold standard key, which will be removed by the MUC scorer if exists but not aligned

4.1 Fixing the Precision Formula

The simple fix on precision calculation (§2.4) does not affect the overall trend of the evaluation. The scores of the participant systems only change by a very small value, and the span-based ordering remains the same. We argue that this fix is both more theoretically sound and mathematically pleasing.

4.2 Combining Multiple Scores

As discussed in §2.6, scoring each metric individually will make it difficult to provide one unified score to rank all systems. This can be seen from Figure 1, which plot the evaluation results using the original scoring (sorted on Span F1). In addition, because attribute scores are only calculated on the gold standard mentions, the false alarms on the rest of the predicted mentions are not penalized.

Figure 2 shows the results using multiplicity combination. We observe that the resulting scores will soon become too small after multiplication, which are less interpretable.

Figure 3 presents the results after applying Algorithm 4. The combined score of all attributes now falls into a more reasonable range (bounded by the performance of the hardest attribute, namely realis status). We also observe that all performances decrease monotonically.

We can also use the results from Figure 3 to understand the performance bottleneck of the systems. For example, in system 7, there is a big gap between the mention type F1 score and the span F1. This indicates that the type detection accuracy is low and should be improved. In system 5, the mention span F1 and mention type F1 are very close. Therefore the bottleneck might be in event span identification. This information is not immediately clear from the other figures.

5 Conclusions

In this paper we describe our proposed evaluation metric for event nugget task and identify two problems in evaluation design. We propose solutions to these problems and find out that the new methods produce more interpretable results.

Acknowledgments

This research was supported in part by DARPA grant FA8750-12-2-0342 funded under the DEFT program.

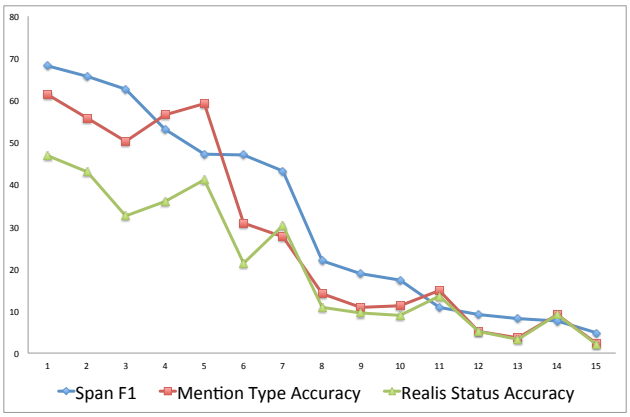


Figure 1: System results sorted by Span F1 score

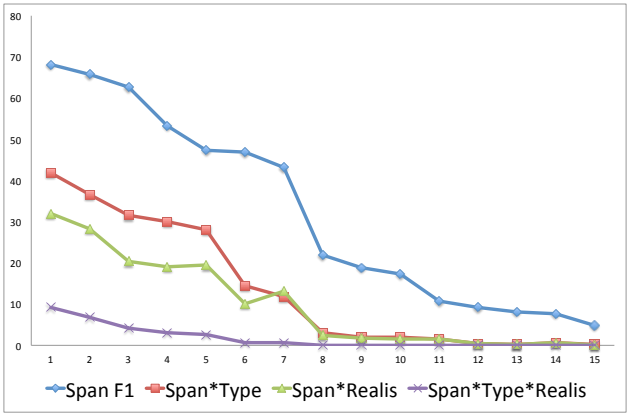


Figure 2: Combining scores with multiplicity (sorted on combined score)

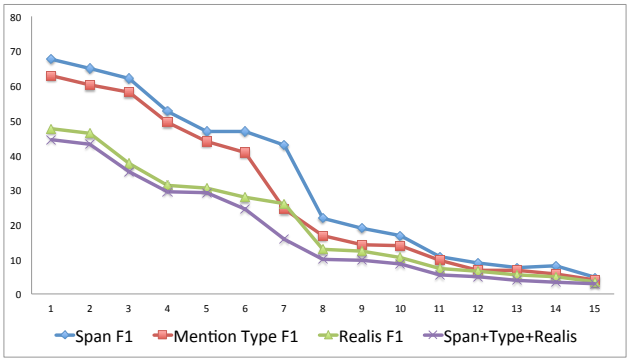


Figure 3: Attribute augmented scoring (sorted on combined score)

References

Cosmin Adrian Bejan, Matthew Titsworth, Andrew Hickl, and Sanda Harabagiu. 2005. Nonparametric Bayesian Models for Unsupervised Event Coreference Resolution. In Y Bengio, D Schuurmans, J Lafferty, C K I Williams, and A Culotta, editors, *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 1–9.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. In *ACL '08 Meeting of the Association for Computational Linguistics*, pages 789–797.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *ACL '09 Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610. Association for Computational Linguistics.

Zheng Chen and H Ji. 2009. Graph-based event coreference resolution. *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 54–57.

Chen Chen and Vincent Ng. 2013. Chinese Event Coreference Resolution: Understanding the State of the Art. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 822–828.

Nancy Chinchor. 1992. Muc-5 evaluation metric. In *Proceedings of the 5th Conference on Message Understanding*, pages 69–78.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint Entity and Event Coreference Resolution across Documents. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint Event Extraction via Structured Prediction with Global Features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL2013)*.

Qi Li, Heng Ji, Yu Hong, and Sujian Li. 2014. Constructing Information Networks Using One Single Model. In *Proceedings the 2014 Conference on Empirical Methods on Natural Language Processing (EMNLP2014)*.

Zhengzhong Liu, Jun Araki, Eduard Hovy, and Teruko Mitamura. 2013. Supervised Within-Document Event Coreference using Information Propagation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4539–4544, Reykjavik, Iceland. European Language Resources Association (ELRA).

Teruko Mitamura. 2014. TAC KBP event detection annotation guidelines, v1.7. Technical report, Carnegie Mellon University, September.

NIST. 2005. The ACE 2005 (ACE05) Evaluation Plan: Evaluation of the Detection and Recognition of ACE. Technical report, National Institute of Standards and Technology.

The ACE 2005 (ACE05) Evaluation Plan

Evaluation of the Detection and Recognition of ACE ***Entities, Values***, ***Temporal Expressions***, ***Relations,*** and ***Events***

1 INTRODUCTION

The objective of the ACE program is to develop automatic content extraction technology to support the automatic processing of source language data. Possible down-stream processing includes classification, filtering, and selection based on the content of the source data, i.e., based on the meaning conveyed by the language. Thus, the ACE program is dedicated to the development of technologies that automatically infer meaning from language data.

2 TASK DEFINITIONS

There are five primary ACE recognition tasks – the recognition of ***entities***, ***values***, ***temporal expressions***, ***relations***, and ***events***. These tasks require systems to process language data in documents and then to output, for each of these documents, information about the entities, values, temporal expressions, relations, and events mentioned or discussed in them. This section provides an overview of the ACE tasks. For a complete description refer to the ACE annotation guidelines.[1] The form of the output that is required is defined by an XML format call "APF". The XML DTD for this format may be obtained from the NIST ACE web site.[2]

In addition to the five primary ACE recognition tasks, this year's ACE evaluation will support three mention-level tasks, namely, *entity mentions*, *relation mentions*, and *event mentions*.

2.1 ENTITY DETECTION AND RECOGNITION

The ACE Entity Detection and Recognition task (EDR) requires that certain specified types of entities that are mentioned in the source language data be detected and that selected information about these entities be recognized and merged into a unified representation for each detected entity. The EDR task will be supported for all three ACE languages, which are ***A***rabic, ***C***hinese and ***E***nglish.

2.1.1 ENTITIES

Entity output is required for each document in which the entity is mentioned. This output includes information about the attributes and mentions of the entity. Entity attributes are currently limited to the entity *type*, the entity *subtype*, the entity *class*, and the *name*(s) used to refer to the entity.

The allowable ACE entity types, subtypes and classes for 2005 are listed in Table 1 and There are no limits on the use of inference and world knowledge in detecting and recognizing entities. The determination should represent the system's best judgment of the source's intention (i.e., the intention of the author or speaker).

Table 2. Entities may have only one type, one subtype and one class. Entity types, subtypes and classes are described in detail in the annotation guidelines. Of the classes discussed in the guidelines, only SPC (specific) entities are assigned a non-zero value during evaluation and therefore systems need output only SPC entities. However, performance on SPC entities may prove to be better if a system attempts to output more than just the SPC entities.

It often happens that different entities may be referred to by the same name. Despite this metonymic connection, however, such entities are regarded as separate and distinct for the purposes of the ACE program. For example, in the sentence "*Miami is growing rapidly*", Miami is a mention of a GPE entity named "Miami", whereas in the sentence "*Miami defeated Atlanta 28 to 3*", Miami is a metonymic mention of an organization entity named "Dolphins" and is distinct from the Miami GPE entity.

Table 1 ACE05 Entity Types and Subtypes

Type	Subtypes
FAC (Facility)	Airport, Building-Grounds, Path, Plant, Subarea-Facility
GPE (Geo-Political Entity[3])	Continent, County-or-District, GPE-Cluster, Nation, Population-Center, Special, State-or-Province
LOC (Location)	Address, Boundary, Celestial, Land-Region-Natural, Region-General, Region-International, Water-Body
ORG (Organization)	Commercial, Educational, Entertainment, Government, Media, Medical-Science, Non-Governmental, Religious, Sports
PER (Person)	Group, Indeterminate, Individual
VEH (Vehicle)	Air, Land, Subarea-Vehicle, Underspecified, Water
WEA (Weapon)	Biological, Blunt, Chemical, Exploding, Nuclear, Projectile, Sharp, Shooting, Underspecified

There are no limits on the use of inference and world knowledge in detecting and recognizing entities. The determination should

[1] http://www.ldc.upenn.edu/Projects/ACE/Annotation/

[2] http://www.nist.gov/speech/tests/ace/ace05/doc/

[3] Geo-Political Entities deserve a little supplemental explanation and historical background. Originally, GPE's were not part of the ACE entity inventory. However, during the initial annotation exercises, it became clear that the same word would often imply different entity types – sometimes *location* (as in "the riots in Miami"), sometimes *organization* (as in "Miami imposed a curfew"), sometimes as *person* (as in "Miami railed against the curfew"). Even more troublesome, co-reference was sometimes observed between different underlying entity types (as in "Miami imposed a curfew because of its riots"). These issues gave rise to the definition of the hybrid Geo-Political entity type. This type can be viewed as somewhat synthetic and ad hoc, but there is also support for its conceptual reality, for example by the use of co-reference in joining different entity types.

represent the system's best judgment of the source's intention (i.e., the intention of the author or speaker).

Table 2 ACE05 Entity Classes

Type	Description
SPC	A particular, specific and unique real world entity
GEN	A kind or type of entity rather than a specific entity
NEG	A negatively quantified (usually generic) entity
USP	An underspecified entity (e.g., modal/uncertain/…)

2.1.2 ENTITY MENTIONS

All mentions of each ACE entity are to be detected and output along with the entity attributes. It is important to output every mention to get full value for each entity. The output for each entity mention includes the mention *type*, the location of its *head* and its *extent*, and optionally the mention *role* and *style* of the mention. Mention *style* is either literal or metonymic. This is currently encoded in the apf file format as an attribute called "metonymy_mention", which is either true (for metonymic style of reference) or false (for literal style of reference). The default style is literal. Mention attributes and their possible values are described in detail in the annotation guidelines. The allowable mention types are listed in Table 3.

Table 3 ACE Mention Types

Type	Description
NAM (Name)	A proper name reference to the entity
NOM (Nominal)	A common noun reference to the entity
PRO (Pronominal)	A pronominal reference to the entity

2.2 VALUE DETECTION AND RECOGNITION

The ACE Value Detection and Recognition task (VAL) requires that certain specified types of values that are mentioned in the source language data be detected and that selected information about these values be recognized and merged into a unified representation for each detected value. The VAL task will be supported for two of the ACE languages (Chinese and English). An ACE value is a quantity that provides additional information and that may also be used, as are entities, as arguments of events. Values are represented similarly to entities and are characterized by their attributes and mentions. The type and subtype attributes of each ACE value for 2005 are listed in Table 4. Value types and subtypes are described in detail in the annotation guidelines.

Table 4 ACE05 Value Types and Subtypes

Type	Subtype
Always annotated when mentioned	
Contact-Info	E-Mail, Phone-Number, URL
Numeric	Money, Percent
Annotated when used as an argument in an Event	
Crime	*none*
Job-Title	*none*
Sentence	*none*

2.3 TIME DETECTION AND RECOGNITION

The ACE Time Expression Recognition and Normalization task (TERN) requires that certain temporal expressions mentioned in the source language data be detected and recognized (in timex2 format) according to the *"TIDES 2005 Standard for the Annotations of Temporal Expressions"* April, 2005[4]. The TERN task will be supported for two of the ACE languages (Chinese and English).

Temporal expressions to be recognized include both absolute expressions and relative expressions. In addition, durations, event-anchored expressions and sets of times are to be recognized. This information is contained in the set of timex2 attributes. The ACE timex2 attributes to be evaluated in 2005 are listed in Table 5.

Table 5 ACE05 timex2 attributes

Attribute	Function
VAL	A normalized time expression
MOD	A normalized time expression modifier
ANCHOR_VAL	A normalized time reference point
ANCHOR_DIR	A normalized time directionality
SET	Designates that VAL is a set of time expressions

Note that this year timex2 elements are being reintroduced as arguments of relations and events. Therefore it is important to recognize them and include them as arguments of relations and events where appropriate.

2.4 RELATION DETECTION AND RECOGNITION

The ACE Relation Detection and Recognition task (RDR) requires that certain specified types of relations that are mentioned in the source language data be detected and that selected information about these relations be recognized and merged into a unified representation for each detected relation. The RDR task will be supported for all three ACE languages.

2.4.1 RELATIONS

An ACE relation is a relation between two ACE entities, which are called the relation arguments. Some relations are symmetric, meaning that the ordering of the two entities does not matter (e.g., "partner"). But for asymmetric relations the order does

[4] See http://timex2.mitre.org for more information regarding definition and annotation of timex2 temporal expressions.

matter (e.g., "subsidiary") and for these relations the entity arguments must be assigned the correct argument role.

Relation output is required for each document in which the relation is mentioned. This output includes information about the attributes of the relation, the relation arguments, and the relation mentions. Relation attributes are the relation *type*, *subtype*, *modality* and *tense*. The ACE relation types and subtypes for 2005 are listed in Table 6. Relations may have only one type and one subtype.

Table 6 ACE05 Relation Types and Subtypes
(Relations marked with an * are symmetric relations.)

Type	Subtype
ART (artifact)	User-Owner-Inventor-Manufacturer
GEN-AFF (Gen-affiliation)	Citizen-Resident-Religion-Ethnicity, Org-Location
METONYMY*	*none*
ORG-AFF (Org-affiliation)	Employment, Founder, Ownership, Student-Alum, Sports-Affiliation, Investor-Shareholder, Membership
PART-WHOLE (part-whole)	Artifact, Geographical, Subsidiary
PER-SOC* (person-social)	Business, Family, Lasting-Personal
PHYS* (physical)	Located, Near

2.4.2 Relation Arguments

Relation arguments are identified by a unique ID and a role. The roles of the two entities being related are "Arg-1" and "Arg-2" and the correct assignment of these roles to their respective arguments is important, except for symmetric relations (which are identified in Table 6). There may be only one Arg-1 entity and one Arg-2 entity. In addition to the two principal entity arguments there may be one or more temporal (timex2) arguments, and it is important to include these arguments in the relation in order to receive full value for the relation. The list of allowable argument roles for relations is given in Table 7.

Table 7 Argument roles allowable for relations

Allowable Relation Roles	
Arg-1	Arg-2
Time-After	Time-Before
Time-At-Beginning	Time-At-End
Time-Starting	Time-Ending
Time-Holds	Time-Within

2.4.3 Relation Mentions

A relation mention is a sentence or phrase that expresses the relation. The extent of the relation mention is defined to be the sentence or phrase within which the relation is mentioned. A relation mention must contain mentions of both of the entities being related. Although recognition of relation mentions is not evaluated, it is one of the ways that system output relations are allowed to map to reference relations. Thus correct recognition of relation mentions is potentially helpful in evaluation.

2.5 Event Detection and Recognition

The ACE Event Detection and Recognition task (VDR) requires that certain specified types of events that are mentioned in the source language data be detected and that selected information about these events be recognized and merged into a unified representation for each detected event. The VDR task will be supported for two ACE languages (Chinese and English).

2.5.1 Events

An ACE event is an event involving zero or more ACE entities, values and time expressions. Event output is required for each document in which the event is mentioned. This output includes information about the attributes of the event, the event arguments, and the event mentions. Event attributes are the event *type*, *subtype*, *modality*, *polarity*, *genericity* and *tense*. The ACE event types and subtypes for 2005 are listed in Table 8. Events may have only one type and one subtype.

Table 8 ACE05 Event Types and Subtypes

Types	Subtype
Life	Be-Born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-Ownership, Transfer-Money
Business	Start-Org, Merge-Org, Declare-Bankruptcy, End-Org
Conflict	Attack, Demonstrate
Contact	Meet, Phone-Write
Personnel	Start-Position, End-Position, Nominate, Elect
Justice	Arrest-Jail, Release-Parole, Trial-Hearing, Charge-Indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

2.5.2 Event Arguments

Each event argument is identified by a unique ID and a role. Unlike relations, which allow only one argument in the Arg-1 and Arg-2 roles, events allow multiple arguments in the same role. The list of allowable argument roles for events is given in Table 9.

Table 9 Argument roles allowable for events

Allowable Event Roles		
Person	Place	Buyer
Seller	Beneficiary	Price
Artifact	Origin	Destination
Giver	Recipient	Money
Org	Agent	Victim
Instrument	Entity Entity	Attacker
Target	Defendant	Adjudicator
Prosecutor	Plaintiff	Crime
Position	Sentence	Vehicle
Time-After	Time-Before	Time-At-Beginning
Time-At-End	Time-Starting	Time-Ending
Time-Holds	Time-Within	

2.5.3 EVENT MENTIONS

An event mention is a sentence or phrase that mentions an event, and the extent of the event mention is defined to be the whole sentence within which the event is mentioned. Although recognition of event mentions is not evaluated, it is one of the ways that system output events are allowed to map to reference events. Thus correct recognition of event mentions is potentially helpful in evaluation.

2.6 ENTITY MENTION DETECTION

The ACE Entity Mention and Detection (EMD) diagnostic task will be supported for all three ACE languages. Section 2.1.2 describes entity mentions.

2.7 RELATION MENTION DETECTION

The ACE Relation Mention and Detection (RMD) diagnostic task will be supported for all three ACE languages. Section 2.4.3 describes relation mentions.

2.8 EVENT MENTION DETECTION

The ACE Event Mention Detection (VMD) diagnostic task will be supported for two of the ACE languages (Chinese and English). Section 2.5.3 describes event mentions.

3 EVALUATION

Evaluation of ACE system performance will be supported for the five primary tasks in three languages. In addition, there will be three diagnostic tasks supported, where partial information is given to the system under test. The evaluation will include several types of sources and one processing mode, as listed in Table 10.

Participation is required on at least one of the primary tasks on at least one of the three languages. For each task/language/mode combination chosen, all source material must be processed by the system being evaluated, including all of the different source types contained in the evaluation data.

Performance on each of the different ACE tasks is measured separately. However, since the arguments of relations and events include ACE entities, values and time expressions, a system's performance on relations and events is strongly affected by the system's underlying performance on these elements.

3.1 EVALUATION METHOD

System performance on each of the several tasks is scored using a model of the application value of system output. This overall value is the sum of the value for each system output entity (or value, time expression, relation or event), accumulated over all system outputs. The value of a system output is computed by comparing its attributes and associated information with the attributes and associated information of the reference that corresponds to it. When system output information differs from that of the reference, value is lost. And when system output is spurious (i.e., there is no corresponding reference), negative value typically results. Perfect system output performance is achieved when the system output matches the reference without error. The overall score of a system is computed as the system output information relative to this perfect output. Detail of the valuation of system output and scoring is given in Appendix A – System Output Value Models.

Historically, it has been found that loss of value is attributable mostly to misses (where a reference has no corresponding system output) and false alarms (where a system output has no corresponding reference). To a lesser extent, value is lost due to errors in determining attributes and other associated information in those cases where the system output actually does have a corresponding reference.

Table 10 ACE05 Evaluation Support

2005 Evaluation			
Primary Evaluation Tasks:	Languages		
	Ara	Chi	Eng
Entity Detection and Recognition (EDR)	✓	✓	✓
Value Detection and Recognition (VAL)		✓	✓
Timex2 Detection and Recognition (TERN)		✓	✓
Relation Detection and Recognition (RDR)	✓	✓	✓
Event Detection and Recognition (VDR)		✓	✓
Entity Mention Detection (EMD)	✓	✓	✓
Relation Mention Detection (RMD)	✓	✓	✓
Event Mention Detection (VMD)		✓	✓
Diagnostic Tasks:			
EDR Co-Reference (given correct mentions)	✓	✓	✓
RDR given correct entities, values and timex2s	✓	✓	✓
VDR given correct entities, values and timex2s		✓	✓
Processing Mode:			
Document-Level	✓	✓	✓
Cross-Document			
Database Reconciled			
Sources:			
Newswire	✓	✓	✓
Broadcast News	✓	✓	✓
Broadcast Conversations			✓
Weblogs	✓	✓	✓
Usenet Newsgroups/Discussion Forum			✓
Conversational Telephone Speech			✓

3.2 EVALUATION TASKS

3.2.1 ENTITY DETECTION AND RECOGNITION (EDR)

The EDR task is to detect (infer) ACE-defined entities from mentions of them in the source language and to recognize and output selected entity attributes and information about these entities, including information about their mentions. Among other things, this requires that all of the mentions of an entity be correctly associated with that entity. The Value of a system output entity is defined as the product of two factors that represent how accurately the entity's attributes are recognized and how accurately the entity's mentions are detected:

$$Value_{sys\_entity} = Entity\_Value(sys\_entity) \cdot Mentions\_Value(\{sys\_mentions\})$$

Refer to appendix A for a complete description of the EDR *Value* formula.

3.2.2 VALUE DETECTION AND RECOGNITION (VAL)

The VAL task is to detect (infer) ACE-defined value elements from mentions of them in the source language and to recognize and output selected value attributes and information, including information about their mentions. While value elements are currently annotated only at the mention level, both their representation and evaluation are done with the same level of abstraction as that used for entities, namely that value elements are globally unique and may have multiple mentions in multiple documents. The evaluation and scoring of value elements is therefore similar to that for entities. Refer to appendix A for a complete description of the VAL *Value* formula.

3.2.3 TIMEX2 DETECTION AND RECOGNITION (TERN)

The TERN task is to detect (infer) ACE-defined timex2 elements from mentions of them in the source language and to recognize and output selected timex2 attributes and information, including information about their mentions. While timex2 elements are currently annotated only at the mention level, both their representation and evaluation are done with the same level of abstraction as that used for entities, namely that timex2 elements are globally unique and may have multiple mentions in multiple documents. The evaluation and scoring of timex2 elements is therefore similar to that for entities. Refer to appendix A for a complete description of the timex2 *Value* formula.

3.2.4 RELATION DETECTION AND RECOGNITION (RDR)

The RDR task is to detect (infer) ACE-defined relations from the source language and to recognize and output selected attributes and information about these relations, including information about their mentions and arguments. A major part of correctly recognizing relations is correctly recognizing the arguments that are related by the relation. Therefore good argument recognition performance is important to achieving good RDR performance. The value of a system output relation is defined as the product of two factors that represent how accurately the relation's attributes are recognized and how accurately the relation's arguments are detected and recognized:

$$Value_{sys\_relation} = Relation\_Value(sys\_relation) \cdot Arguments\_Value(\{sys\_arguments\})$$

Refer to appendix A for a complete description of the RDR *Value* formula.

3.2.5 EVENT DETECTION AND RECOGNITION (VDR)

The VDR task is to detect (infer) ACE-defined events from the source language and to recognize and output selected attributes and information about these events, including information about their mentions and arguments. A major part of correctly recognizing events is correctly recognizing the arguments that participate in the event. Therefore good argument recognition performance is important to achieving good VDR performance. The value of a system output event is defined as the product of two factors that represent how accurately the event's attributes are recognized and how accurately the event's arguments are detected and recognized:

$$Value_{sys\_event} = Event\_Value(sys\_event) \cdot Arguments\_Value(\{sys\_arguments\})$$

Refer to appendix A for a complete description of the VDR *Value* formula.

3.2.6 ENTITY MENTION DETECTION (EMD)

The EMD task is to detect (infer) all mentions of ACE-defined entities in the source language and to recognize and output selected attributes and information about these entity mentions. Unlike EDR, EMD does not require that mentions of an entity be correctly associated with an entity. Nevertheless, co-reference remains an important issue because each entity mention must be a mention of an entity within the set of ACE entities.

The EMD value formula is identical to that for EDR. For EMD, however, each entity mention is promoted to "entity" status, separately from other mentions, and thus becomes an entity with only one mention.

3.2.7 RELATION MENTION DETECTION (RMD)

RMD is a derivative task that supports diagnostic evaluation of relation mentions. In RMD, each relation mention, for both system output and reference relations, is promoted to "relation" status and becomes a separate and independent relation and is then evaluated as in RDR. There are several differences between mapping and scoring for RMD and RDR, however. This stems from an inherent ambiguity in specifying the mentions of relation arguments, because often times there are several possible choices. This ambiguity is handled in the following way:

- System output argument mentions are promoted to separate independent argument elements (including entities, values and times). Reference argument mentions are not promoted and are left unchanged as mentions of larger elements. This allows a system argument mention to map to any of the reference argument mentions.

Two other differences between RMD and RDR scoring provide the desired RMD score characteristics:

- Positive overlap is required between reference and system output "extents", defined as the span of their Arg-1/Arg-2 mention heads.
- Argument values are defined to be 1 if the arguments are mappable, 0 otherwise. (A system argument is "mappable" if it has a non-null score with the corresponding reference argument.)

3.2.8 EVENT MENTION DETECTION (VMD)

VMD is a derivative task that supports diagnostic evaluation of event mentions. In VMD, each event mention, for both system output and reference events, is promoted to "event" status and becomes a separate and independent event and is then evaluated as in VDR. There are several differences between mapping and scoring for VMD and VDR, however. This stems from an inherent ambiguity in specifying the mentions of event arguments, because often times there are several possible choices. This ambiguity is handled in the following way:

- System output argument mentions are promoted to separate independent argument elements (including entities, values and times). Reference argument mentions are not promoted and are left unchanged as mentions of larger elements. This allows a system argument mention to map to any of the reference argument mentions.

Two other differences between VMD and VDR scoring provide the desired VMD score characteristics:

- Positive overlap is required between reference and system output mention extents.
- Argument values are defined to be 1 if the arguments are mappable, 0 otherwise. (A system argument is "mappable" if it has a non-null score with the corresponding reference argument.)

3.3 CORPUS SUPPORT

Source language data is being provided to support research (with training corpora that may be subdivided to include a development test set) and evaluation (with an evaluation test corpus). ACE corpora are assembled from a variety of sources selected from broadcast news programs, newspapers, newswire reports, internet sources, and from transcribed audio.

3.3.1 THE ACE 2005 TRAINING CORPUS

The Linguistic Data Consortium has newly annotated ACE training data available[5] for system development. The data is taken from a variety of sources and is available for tasks in all three ACE languages: Arabic, Chinese and English.

ACE05 training and evaluation data was selected using a careful targeted process. Rather than choosing files at random for annotation, as was done in past ACE evaluations, this year's task required a certain density of annotation across the corpus

ACE training corpus statistics including publishing dates are listed in Table 11.

Four versions of each document are provided:

- Source text files (.sgm): All source files, including the Chinese files, are encoded in UTF-8. These files use the UNIX-style end of lines. Only text between the begin text tag <TEXT> and end text tag </TEXT> are to be evaluated. The one exception to this rule is that one TIMEX2 annotation is placed between the <DATETIME> and </DATETIME> tags even though they occur outside the TEXT tags.
- APF files (.apf.xml): The ACE Program Format[6].
- AG files (.ag.xml): The LDC Annotation Graph Format. LDC's internal annotation files format for ACE. These files can be viewed with LDC's annotation tool.
- TABLE files (.tab): Files that store mapping tables between the IDs used in each ag.xml file and their corresponding apf.xml file.

To verify data format integrity, three DTD's are distributed with the ACE training corpus. One DTD is used to verify the APF format, one to verify the AG format, and one to verify the original source document format.

[5] Registered participants will be contacted by the LDC with instructions on how to obtain the ACE 2005 training corpus (LDC2005E18).

[6] The ACE APF format is defined by the DTD located at: http://www.nist.gov/speech/tests/ace/ace05/doc/

Table 11 2005 ACE system training corpus statistics for release LDC2005E18. This will be an incremental release. Numbers shown represents total size of final release.

Source	Training epoch	Approximate size
English Resources		
Broadcast News	3/03 – 6/03	60,000 words
Broadcast Conversations	3/03 – 6/03	45,000 words
Newswire	3/03 – 6/03	60,000 words
Weblog	11/04 – 2/05	45,000 words
Usenet	11/04 – 2/05	45,000 words
Conversational Telephone Speech	11/04-12/04 (differentiated by topic vs. eval)	45,000 words
Arabic Resources		
Broadcast News	10/00 – 12/00	60,000 words
Newswire	10/00 – 12/00	60,000 words
Weblog	11/04 – 2/05	30,000 words
Chinese Resources *(1.5 characters = 1 word)*		
Broadcast News	10/00 – 12/00	120,000 words
Newswire	10/00 – 12/00	120,000 words
Weblog	11/04 – 2/05	60,000 words

3.3.2 THE 2005 EVALUATION CORPUS

A new evaluation data set is defined for the 2005 evaluation. Table 12 lists the statistics, including the publication dates, of the ACE05 evaluation corpus.

A key part of system output is the specification of entity mentions in terms of word locations in the source text. Word/phrase location information is in terms of the indices of the first and last characters of the word/phrase. ACE systems must compute these indices from the source data. Indices start with index 0 being assigned to the first character of a document. Ancillary information and annotation, which is provided as bracketed SGML tags, is not included in this count. Only characters (including white-spaces) outside of angle-bracketed expressions contribute to the character count. Also, each new line (nl or cr/lf) counts as one character.

Table 12 The ACE05 evaluation corpus statistics.

Source	Test epoch	Approximate size
English Resources		
Broadcast News	7/03 – 8/03	10,000 words
Broadcast Conversations	7/03 – 8/03	7,500 words
Newswire	7/03 – 8/03	10,000 words
Weblog	3/05 – 4/05	7,500 words
Usenet	3/05 – 4/05	7,500 words
Conversational Telephone Speech	11/04 – 12/04 (different topics from training)	7,500 words
Arabic Resources		
Broadcast News	1/01	20,000 words
Newswire	1/01	20,000 words
Weblog	3/05 – 4/05	10,000 words
Chinese Resources *(1.5 characters = 1 word)*		
Broadcast News	1/01	20,000 words
Newswire	1/01	20,000 words
Weblog	3/05 – 4/05	10,000 words

3.3.3 2005 EVALUATION AND SCORING CONDITIONS

All scoring will be done at the document level. This means that each ACE target (entity, time expression, relation, event or value) will contribute to the score for each document that mentions that target. For example, if an entity is mentioned in N different documents, that entity will contribute to the score N times.

All ACE05 tasks will be scored using "document-level processing" mode.

Document-level processing. For this processing mode, each document is processed independently of other documents. No reconciliation ACE targets are required (or allowed), either across documents or with respect to a database. Thus all entities and relations mentioned in a single document must be uniquely associated and identified with that document. This means, by way of example, that if a specific person, say the US president George W. Bush, is mentioned in more than one document, then he must be represented by multiple entities – a different entity (with a globally unique ID) for each document in which he is mentioned.

There are different source conditions depending on the language of the task. Scores will be reported over the entire evaluation test set as well as separately for each source domain. This will support contrasts between different sources.

3.4 RULES

- No changes to the system are allowed once the evaluation data are released. Adaptive systems may of course change themselves in response to the source data that they process.

- No human intervention is allowed prior to the submission of your test site's results to NIST.[7] This means that, in addition to disallowing modifications to your system, there must also be no modifications to, or human examination of, the test data.
- For each evaluation combination of task, language, and processing mode for which system output is submitted, all documents from all sources for that evaluation combination must be processed.
- NIST will email the evaluation test data to each site on 11/09/05. Sites must return results to NIST within a 24 hour period. The actual starting time on 11/09 is negotiable[8].
- Every participating site must submit a detailed system description to NIST by 11/30/05, as defined in section 3.7.2.
- Every participating site must attend the evaluation workshop and present a system talk if requested[9].

3.5 TOOLS

3.5.1 XML VALIDATION TOOLS

A java implementation of an XML validator[10] is available from the NIST ACE web site. The XML validator will verify that a system output file conforms to the current ACE DTD.[11]

Before sites submit their system results to NIST for scoring, they must validate the results file using the XML validation tool and the current ACE APF DTD. ***Results that are not validated will not be accepted.***

3.5.2 ACE EVALUATION SOFTWARE

The ACE evaluation software is available for download from the NIST ACE web site.[12] This tool scores EDR, VAL, TERN, RDR, and VDR output.

[7] It sometimes happens that a system bug is discovered during the course of processing the test data. In such a case, please consult with NIST email (ace_poc@nist.gov) for advice. NIST will advise you on how to proceed. Repairs may be possible that allow a more accurate assessment of the underlying performance of a system. If this happens, modified results may be accepted, provided that an explanation of the modification is provided and provided that the original results are also submitted and documented.

[8] By default, NIST will send the evaluation data to the registered participants at 9:00am EST, with results due back 24 hours later. It may be desirable for some sites to receive the data at some other time on 11/09/05. It is the registered sites responsibility to contact NIST (ace_poc@nist.gov) to schedule the exact time of data delivery.

[9] Note, not all participants will be requested to give a site talk. The workshop will include a poster session where everyone will have the opportunity to discuss their work. The number of site talks will depend on the number of participants, the innovations of their system algorithms, the tasks and languages attempted, and the quality of the results.

[10] URL: http://www.nist.gov/speech/tests/ace/ace05/software.htm

[11] The DTD's used for the ACE program, can be found at: http://www.nist.gov/speech/tests/ace/dtd/

[12] The ACE evaluation tools may be accessed from the NIST ACE URL http://www.nist.gov/speech/tests/ace/ace05/software.htm

3.6 SCHEDULE

Table 13 The ACE 2005 Evaluation Schedule

Date	Event
11/01/05	Deadline to register[13] for participation in the ACE05 evaluation.
11/07/05	ACE05 Arabic evaluation day
11/08/05	ACE05 Chinese evaluation day
11/09/05	ACE05 English evaluation day
11/14/05	Ground-truth entity mentions available for diagnostic EDR task
11/16/05	(noon deadline, EST) Diagnostic EDR results due at NIST
11/16/05	Ground-truth ENTITIES available for diagnostic RDR and VDR tasks
11/18/05	(COB deadline) Diagnostic RDR and VDR results due at NIST
11/23/05	NIST releases results
11/30/05	(noon deadline, EST) Site's detailed system description papers are due at NIST
12/05/05	A handful of sites will receive requests to give formal talks at the evaluation workshop.
12/15-16/05	Two day evaluation workshop.

3.7 SUBMISSION OF SYSTEM OUTPUT TO NIST

To enable quick unpacking and scoring of several site submission files with minimum human intervention, participants must follow the outlined procedure for submitting results.

3.7.1 PACKAGING YOUR SYSTEM OUTPUT

Note, that in many cases a system output file will contain results for more than one task (i.e. EDR and RDR). In such a case the exact same set of files should be copied to the EDR and RDR subdirectories as defined below.

STEP1: Create a top level directory for each of the ***languages*** attempted (Arabic | Chinese | English):

Example: $> mkdir chinese english

STEP2: Create a subdirectory identifying the ***tasks*** attempted (EDR | VAL | TERN | RDR| VDR):

Example: $> mkdir english/edr english/rdr chinese/edr

STEP3: In each of these subdirectories make one directory for each system submitted (choose a name that identifies your site, BBN, SHEF, SRI…):

Example: $> mkdir english/edr/NIST1_primary

Example: $> mkdir english/edr/NIST2_contrastive1

Example: $> mkdir english/rdr/NIST1_primary

[13] The official ACE05 registration form is located at the URL: http://www.nist.gov/speech/tests/ace/ace05/doc/

Example: $> mkdir chinese/edr/NIST1_primary

STEP4: Deposit all system output files in the appropriate system directory.

STEP5: Create a compressed tar file of your results and transfer them to NIST by FTP (ftp://ijaguar.ncsl.nist.gov/incoming). After successful transmission send e-mail to ace_poc@nist.gov identifying the name of the file submitted. Alternatively you may send the compressed tar file directly to ace_poc@nist.gov .

3.7.2 SYSTEM DESCRIPTION

A valuable tool in discovering strengths and weakness of different algorithmic approaches is the use of system descriptions. This year, system descriptions will also be used to help determine which sites are to give oral workshop presentations and which sites are to give talks in a poster session.

Each participant must prepare a ***detailed*** system description covering each system submitted. System descriptions are due at NIST no later than 11/30/05. It is important that all sites submit comprehensive descriptions on time so that NIST may plan the workshop agenda accordingly.

These system descriptions will be distributed to each participant before the evaluation workshop.

Each system description should include:

- The ACE tasks and languages processed
- Identification of the primary system for each task
- A description of the system (algorithms, data, configuration) used to produce the system output
- How contrastive systems differ from the primary system
- A description of the resources required to process the test set, including CPU time and memory
- Applicable references

4 GUIDELINES FOR PUBLICATIONS

NIST Speech Group's HLT evaluations are moving towards an open model which promotes interchange with the outside world. Therefore, the rules governing the publication of ACE05 evaluation results have been updated..

4.1 NIST PUBLICATION OF RESULTS

At the conclusion of the evaluation cycle, NIST will create a report which documents the evaluation. The report will be posted on the NIST web space and will identify the participants and official ACE value scores achieved for each task/language combination. Scores will be reported for the overall test set and for the different data sources.

The report that NIST creates should not be construed, or represented as endorsements for any participant's system or commercial product, or as official findings on the part of NIST or the U.S. Government.

4.2 PARTICIPANT'S PUBLICATION OF RESULTS

Participants will be free to publish results for their own system, and may state the highest score achieved in a particular task, but sites will not be allowed to name other participants, or cite another site's results without permission from the other site. Publications may point to the NIST report as a reference[14].

[14] This restriction exists to ensure that readers concerned with a particular system's performance will see the entire set of participants and tasks attempted by all researchers.

APPENDIX A – SYSTEM OUTPUT VALUE MODELS

EDR SCORING

The EDR value score for a system is defined to be the sum of the values of all of the system's output entity tokens, normalized by the sum of the values of all reference entity tokens. The maximum possible EDR value score is 100 percent.

$$EDR\_Value_{sys} = \sum_i value\_of\_sys\_token_i \Big/ \sum_j value\_of\_ref\_token_j$$

The value of each system token is based on its attributes and on how well it matches its corresponding reference token. A globally optimum correspondence between system and reference tokens which maximizes *EDR_Value* is determined and used, subject to the constraint of one-to-one mapping between system and reference tokens.[15] The value of a system token is defined as the product of two factors that represent both the inherent value of the token and how accurately the token's attributes are recognized and the token's mentions are detected.

$$Value(token) = Element\_Value(token) \cdot Mentions\_Value(token)$$

Element_Value is a function of the attributes of the system token and, if mapped, how well they match those of the corresponding reference token. The inherent value of a token is defined as the product of the token's attribute value parameters, *AttrValue*, for the attributes ***type*** and ***class***. This inherent value is reduced for any attribute errors (i.e., for any differences between the values of system and reference attributes), using the error weighting parameters, $\{W_{err\text{-}attribute}\}$. If a system token is unmapped, then the value of that token is weighted by a false alarm penalty, $W_{E\text{-}FA}$.

$$Element\_Value(sys) = \begin{cases} \min\left(\prod_{\substack{attribute= \\ type,class}} AttrValue(attribute_{sys}), \prod_{\substack{attribute= \\ type,class}} AttrValue(attribute_{ref}) \right) \cdot \prod_{\substack{attribute= \\ type,subtype,class}} W_{err-attribute} & \text{if } sys \text{ mapped} \\ \left(\prod_{\substack{attribute= \\ type,class}} AttrValue(attribute_{sys}) \right) \cdot W_{FA} & \text{if not mapped} \end{cases}$$

Mentions_Value is a function of the mutual mention value (*MMV*) between the mentions of the system token and, if mapped, those of the corresponding reference token[16]. A mention's *MMV* depends on the mention's type value parameter, *MTypeValue*, with this value being reduced for any mention attribute errors (i.e., for any differences between the attribute values of system and reference mentions), W_{Merr}. If the system mention is unmapped, then the *MMV* is weighted by a false alarm penalty factor, $W_{M\text{-}FA}$, and also by a co-reference weighting factor, $W_{M\text{-}CR}$, if the system mention happens to correspond to a legitimate reference mention but one that doesn't belong to the corresponding reference token[17].

$$MMV(mention_{sys}) = \begin{cases} \min\left(MTypeValue(mention_{sys}), MTypeValue(mention_{ref}) \right) \cdot \prod_{\substack{attribute= \\ type,role,style}} W_{Merr-attribute} & \text{if } mention_{sys} \text{ mapped} \\ -MTypeValue(mention_{sys}) \cdot (W_{M-FA} \cdot W_{M-CR}) & \text{if not mapped} \end{cases}$$

For each pairing of a system token with a reference token, an optimum correspondence between system mentions and reference mentions that maximizes the sum of *MMV* over all system mentions is determined and used, subject to the constraint of one-to-one mapping between system and reference mentions.

Mentions_Value is computed using one of two formulas, depending on whether valuation is ***mention***-weighted or ***level***-weighted. For mention-weighted valuation *Mentions_Value* is simply the sum of *MMV* over all mentions in all documents. For level-weighted valuation *Mentions_Value* is determined by a system token's "level" (and that of its corresponding reference token) and by the number of documents in which the token is mentioned. The "level" of a token is the highest (i.e., the most valued) mention type of that token. Thus, for example, the "level" of a token is NAM (named) if any one of its mentions is of type NAM, because NAM mentions are more

[15] System tokens and reference tokens are permitted to correspond only if they each have at least one mention in correspondence with the other.

[16] All mentions of a system token are considered to be unmapped for tokens that are themselves unmapped. Thus, for tokens that are unmapped, *Mentions_Value* will be negative. (Note the minus sign in the formula for the *MMV* of unmapped mentions.)

[17] This is intended to avoid double penalizing co-reference errors, namely once for missing the mention in the correct token and once for including the mention in the wrong token. Setting $W_{M\text{-}CR}$ to zero eliminates the second penalty.

valuable than NOM mentions. If none of its mentions is of type NAM, but at least one mention is of type NOM, then the "level" of that token would be NOM (nominal).

$$Mentions\_Value(sys) = \begin{cases} \sum_{\substack{all\\docs}} \left(\sum_{\substack{all\ sys\\mentions\\in\ doc}} MMV(m_{sys}) \right) & \text{if mention - weighted} \\ \min\left(\begin{matrix} MTypeValue(level_{sys}) \\ MTypeValue(level_{ref}) \end{matrix} \right) \cdot \sum_{\substack{all\\docs}} \left(\sum_{\substack{all\ sys\\mentions\\in\ doc}} MMV(m_{sys}) \Big/ \sum_{\substack{all\ ref\\mentions\\in\ doc}} MMV(m_{ref}) \right) & \text{if level - weighted} \end{cases}$$

System mentions and reference mentions are permitted to correspond only if their **heads** have a mutual overlap of at least *min_overlap* and the text of their **heads** share a (fractional) consecutive string of characters[18] of at least *min_text_match*. Mention regions and overlaps are measured in terms of *number of characters* for text input, in terms of *time* for audio input, and in terms of *area* for image input.

$$mutual\_overlap = \frac{sys\_head \cap ref\_head}{\max(sys\_head, ref\_head)}$$

$$fractional\_consecutive\_string = \frac{\left(\begin{matrix} \#\ of\ characters\ in\ the\ longest\ consecutive\ string\ of\ characters \\ that\ is\ contained\ in\ both\ system\ and\ reference\ mention\ head\ texts \end{matrix} \right)}{\max\left(\begin{matrix} \#\ of\ characters\ in\ system\ mention\ head\ text, \\ \#\ of\ characters\ in\ reference\ mention\ head\ text \end{matrix} \right)}$$

The current default scoring parameters for EDR are given in Table 14.

Table 14 Default parameters for scoring EDR performance

***Element_Value* parameters**			
Attribute	$W_{err\text{-}attribute}$	**Attribute Value**	***AttrValue***
Type	0.50	(all types)	1.00
Class	0.75	SPC	1.00
		(not SPC)	0.00
Subtype	0.90	n/a	n/a
$W_{E\text{-}FA}$ = 0.75			
***Mentions_Value* parameters**			
Attribute	$W_{Merr\text{-}attribute}$	**Attribute Value**	***MTypeValue***
Type	0.90	NAM	1.00
		NOM	0.50
		PRO	0.10
Role	0.90	n/a	n/a
Style	0.90	n/a	n/a
Valuation = level-weighted			
$W_{M\text{-}FA}$ = 0.75		$W_{M\text{-}CR}$ = 0.00	
min_overlap = 0.30		***min_text_match*** = 0.30	

[18] This requirement of a common substring in both system and output mention heads was invoked to account for errors in transcribing speech and image data into text. The intent is to require a mention be meaningful and relevant in order to be counted.

VAL SCORING

The VAL value score for a system is defined to be the sum of the values of all of the system's output value tokens, normalized by the sum of the values of all of the reference value tokens. The maximum possible VAL value score is 100 percent.

$$VAL\_Value_{sys} = \sum_i value\_of\_sys\_token_i \Big/ \sum_j value\_of\_ref\_token_j$$

The value of each system token is based on its attributes and on how well it matches its corresponding reference token. A globally optimum correspondence between system and reference tokens which maximizes *VAL_Value* is determined and used, subject to the constraint of one-to-one mapping between system and reference tokens.[15] The value of a system token is defined as the product of two factors that represent both the inherent value of the token and how accurately the token's attributes are recognized and the token's mentions are detected.

$$Value(token) = Element\_Value(token) \cdot Mentions\_Value(token)$$

Element_Value depends on the token type and, if mapped, on how well the attributes of the system token match those of the corresponding reference token. The inherent value of a token is determined by the token's type value parameter, *AttrValue*(*type*). This inherent value is reduced for any attribute errors (i.e., for any differences between the values of system and reference attributes), using the error weighting parameters, $\{W_{err\text{-}attribute}\}$. If a system token is unmapped, then the value of that token is weighted by a false alarm penalty, W_{FA}.

$$Element\_Value(sys) = \begin{cases} \min\begin{pmatrix} AttrValue(type_{sys}), \\ AttrValue(type_{ref}) \end{pmatrix} \cdot \prod_{\substack{attribute= \\ type,subtype}} W_{err-attribute} & \text{if } sys \text{ mapped} \\ \\ \left(AttrValue(type_{sys})\right) \cdot W_{FA} & \text{if not mapped} \end{cases}$$

Mentions_Value is simply the sum of the mutual mention value (*MMV*) between the mentions of the system token and, if mapped, those of the corresponding reference token[16]. A mention's *MMV* is simply the value 1. If the system mention is unmapped, then the *MMV* is weighted by a false alarm penalty factor, $W_{M\text{-}FA}$, and also by a co-reference weighting factor, $W_{M\text{-}CR}$, if the system mention happens to correspond to a legitimate reference mention but one that doesn't belong to the corresponding reference token[17]. For each pairing of a system token and a reference token, an optimum correspondence between system mentions and reference mentions that maximizes the sum of *MMV* over all system mentions is determined and used, subject to the constraint of one-to-one mapping between system and reference mentions.

$$MMV(mention_{sys}) = \begin{cases} 1 & \text{if } mention_{sys} \text{ mapped} \\ \\ -(W_{M-FA} \cdot W_{M-CR}) & \text{if not mapped} \end{cases} \qquad Mentions\_Value(sys) = \sum_{\substack{all \\ docs}} \left(\sum_{\substack{all\ sys \\ mentions \\ in\ doc}} MMV(m_{sys}) \right)$$

System mentions and reference mentions are permitted to correspond only if their **extents** have a mutual overlap of at least *min_overlap*. Mention regions and overlaps are measured in terms of *number of characters* for text input, in terms of *time* for audio input, and in terms of *area* for image input.

$$mutual\_overlap = \frac{sys\_extent \cap ref\_extent}{\max(sys\_extent, ref\_extent)}$$

The current default parameters for VAL scoring are given in Table 15.

Table 15 Default parameters for scoring VAL performance

***Element_Value* parameters**				***Mentions_Value* parameters**	
Attribute	$W_{err\text{-}attribute}$	**Attribute Value**	***AttrValue***	$W_{Merr\text{-}attribute}$	0.90 (for all attributes)
Type	0.50	(all types)	1.00	$W_{M\text{-}FA}$	0.75
Subtype	0.90	n/a	n/a	$W_{M\text{-}CR}$	0.00
W_{FA} = 0.75				***min_overlap***	0.30

TERN SCORING

The TERN value score for a system is defined to be the sum of the values of all of the system's output timex2 tokens, normalized by the sum of the values of all of the reference timex2 tokens. The maximum possible timex2 value score is 100 percent.

$$TERN\_Value_{sys} = \sum_i value\_of\_sys\_token_i \Big/ \sum_j value\_of\_ref\_token_j$$

The value of each system token is based on its attributes and on how well it matches its corresponding reference token. A globally optimum correspondence between system and reference tokens which maximizes *TERN_Value* is determined and used, subject to the constraint of one-to-one mapping between system and reference tokens.[15] The value of a system token is defined as the product of two factors that represent both the inherent value of the token and how accurately the token's attributes are recognized and the token's mentions are detected.

$$Value(token) = Element\_Value(token) \cdot Mentions\_Value(token)$$

Element_Value depends on how well the attributes of the system token match those of the corresponding reference token. The inherent value of a token is defined as a sum of attribute value parameters, *AttrValue*, summed over all attributes which exist and which are the same for both the system and reference tokens. If a system token is unmapped, then the value of that token is weighted by a false alarm penalty, W_{FA}.

$$Element\_Value(sys) = \begin{cases} \sum_{\substack{\text{for all existing } sys \text{ attributes in the set} \\ \{type, mod, set, val, anchor\_dir, anchor\_val\}}} \begin{cases} AttrValue(attribute) & \text{if } attribute_{sys} = attribute_{ref} \\ 0 & \text{otherwise} \end{cases} & \text{if } sys \text{ mapped} \\ \sum_{\substack{\text{for all existing } sys \text{ attributes in the set} \\ \{type, mod, set, val, anchor\_dir, anchor\_val\}}} AttrValue(attribute) \cdot W_{FA} & \text{if not mapped} \end{cases}$$

Mentions_Value is simply the sum of the mutual mention value (*MMV*) between the mentions of the system token and, if mapped, those of the corresponding reference token[16]. A mention's *MMV* is simply the value 1. If the system mention is unmapped, then the *MMV* is weighted by a false alarm penalty factor, $W_{M\text{-}FA}$, and also by a co-reference weighting factor, $W_{M\text{-}CR}$, if the system mention happens to correspond to a legitimate reference mention but one that doesn't belong to the corresponding reference token[17]. For each pairing of a system token and a reference token, an optimum correspondence between system mentions and reference mentions that maximizes the sum of *MMV* over all system mentions is determined and used, subject to the constraint of one-to-one mapping between system and reference mentions.

$$MMV(mention_{sys}) = \begin{cases} 1 & \text{if } mention_{sys} \text{ mapped} \\ -(W_{M-FA} \cdot W_{M-CR}) & \text{if not mapped} \end{cases} \qquad Mentions\_Value(sys) = \sum_{\substack{all \\ docs}} \left(\sum_{\substack{all\ sys \\ mentions \\ in\ doc}} MMV(m_{sys}) \right)$$

System mentions and reference mentions are permitted to correspond only if their **extents** have a mutual overlap of at least *min_overlap*. Mention regions and overlaps are measured in terms of *number of characters* for text input, in terms of *time* for audio input, and in terms of *area* for image input.

$$mutual\_overlap = \frac{sys\_extent \cap ref\_extent}{\max(sys\_extent, ref\_extent)}$$

The current default parameters for TERN scoring are given in Table 16.

Table 16 Default parameters for scoring TERN performance

***Element_Value* parameters**						
attribute	type	anchor_dir	anchor_val	mod	set	val
AttrValue	0.10	0.25	0.50	0.10	0.10	1.00
$W_{E\text{-}FA} = 0.75$						
***Mentions_Value* parameters**						
$W_{M\text{-}FA} = 0.75$		$W_{M\text{-}CR} = 0.00$			***min_overlap*** = 0.30	

RDR SCORING

The RDR value score for a system is defined to be the sum of the values of all of the system's output relation tokens, normalized by the sum of the values of all reference relation tokens. The maximum possible RDR value score is 100 percent.

$$RDR\_Value_{sys} \;=\; \sum_i value\_of\_sys\_token_i \;\Big/\; \sum_j value\_of\_ref\_token_j$$

The value of each system token is based on its attributes and arguments and on how well they match those of a corresponding reference token. A globally optimum correspondence between system and reference tokens which maximizes *RDR_Value* is determined and used, subject to the constraint of one-to-one mapping between system and reference tokens. System tokens and reference tokens are permitted to correspond only if they have some nominal basis for correspondence. The required nominal basis is selectable from the set of minimal conditions listed in Table 17.

Table 17 Conditions required for correspondence between system and reference relation tokens

Condition	Description
arguments	At least one argument in the system token must be mappable to an argument in the reference token.
extents	The system and reference tokens must each have at least one mention extent in correspondence with the other.
both	Both the **arguments** condition and the **extents** condition must be met.
either	Either the **arguments** condition or the **extents** condition must be met.
all	All arguments in the reference token must be one-to-one mappable to arguments in the system token.
all+extents	Both the **all** condition and the **extents** condition must be met.

The value of a system token is defined as the product of two factors that represent both the inherent value of the token and how accurately the token's attributes and arguments are recognized.

$$Value(token) \;=\; Element\_Value(token) \;\cdot\; Arguments\_Value(token)$$

Element_Value is a function of the attributes of the system token and, if mapped, how well they match those of the corresponding reference token. The inherent value of a token is defined as the product of the token's attribute value parameters, *AttrValue*, for the attributes ***type*** and ***modality***. This inherent value is reduced for any attribute errors (i.e., for any difference between the values of system and reference attributes), using the error weighting parameters, $\{W_{err\text{-}attribute}\}$. If a system token is unmapped, then the value of that token is weighted by a false alarm penalty, W_{FA}.

$$Element\_Value(sys) \;=\; \begin{cases} \min\left(\prod_{\substack{attribute= \\ type,modality}} AttrValue(attribute_{sys}), \prod_{\substack{attribute= \\ type,modality}} AttrValue(attribute_{ref}) \right) \cdot \prod_{\substack{attribute= \\ type,subtype,modality,tense}} W_{err-attribute} & \text{if mapped} \\ AttrValue(attribute_{sys}) \cdot W_{FA} & \text{if not mapped} \end{cases}$$

Arguments_Value is a function of the mutual argument value (*MAV*) between the arguments of the system token and, if mapped, those of the corresponding reference token.[19] An argument's *MAV* depends on the system argument's value (with respect to the putative reference argument) for each document in which the relation is mentioned, $Value_{doc}(arg_{sys},arg_{ref})$, with this value being reduced for argument role errors (i.e., for a difference between the roles of system and reference arguments), $W_{err\text{-}role}$. Argument-level errors are accounted for using an incremental formulation of false alarm error. Specifically, loss of value at the argument level is viewed as a partial false alarm, and this loss of value is subtracted from the *MAV* after being weighted by a false alarm penalty factor, $W_{A\text{-}FA}$.

$$MAV_{doc}(arg_{sys}) \;=\; Value_{doc}(arg_{sys}, arg_{ref}) \cdot W_{err-role} \;-\; \left(Value_{doc}(arg_{sys}, arg_{sys}) - Value_{doc}(arg_{sys}, arg_{ref})\right) \cdot W_{A-FA}$$

[19] All arguments of a system token are considered to be unmapped for tokens that are themselves unmapped. Thus, for tokens that are unmapped, *Arguments_Value* will be negative. Note that *MAV* is negative for unmapped arguments, i.e., when $Value_{doc}(arg_{sys},arg_{ref}) = 0$.

If there is no corresponding reference argument for a system argument, then $Value_{doc}(arg_{sys},arg_{ref})$ is taken to be zero. There are several requirements that must be satisfied in order for a reference argument to be considered to be in correspondence to a system argument. First, note that there are two required arguments, namely the two arguments for which the relation is being asserted. These arguments have roles called "Arg-1" and "Arg-2", and there may be only one Arg-1 and one Arg-2 argument.[20] The requirements for correspondence are listed in Table 18.

Table 18 Conditions required for correspondence between system and reference relation arguments

Condition	Requirement
Always	The reference argument must be mappable to the system argument. That is, they must have at least one mention in correspondence.
If the "**mapped**" argument option is invoked	The reference argument must correspond to the system argument. That is, they must be mapped to each other at the argument level.
Argument role is Arg-1 or Arg-2 and the relation symmetric	The reference argument role may be either "Arg-1" or "Arg-2".
Argument role is Arg-1 or Arg-2 and the relation is **not** symmetric	The reference argument role may be either "Arg-1" or "Arg-2", but *Element_Value* is reduced if ref and sys roles do not match.[21]

For each pairing of a system relation token with a reference relation token, an optimum correspondence between system arguments and reference arguments that maximizes *Arguments_Value* is determined and used. This optimum mapping is constrained to be a one-to-one mapping between system and reference arguments.

Arguments_Value is computed using one of two formulas, depending on whether the contribution of the various relation arguments are averaged arithmetically or geometrically.

$$Arguments\_Value(sys) = \begin{cases} \sum_{all\ arg_{sys}} \left(\sum_{\substack{all\ docs\ that \\ mention\ the\ relation}} MAV_{doc}\left(arg_{sys},arg_{ref}\right) \right) & \text{if arithmetic averaging} \\ \prod_{all\ arg_{sys}} \left(\sum_{\substack{all\ docs\ that \\ mention\ the\ relation}} MAV_{doc}\left(arg_{sys},arg_{ref}\right) \right) & \text{if geometric averaging} \end{cases}$$

Note that geometric averaging is sensible only when the *MAV* value contributions exist and are positive for all reference arguments. Thus, for geometric averaging, all reference arguments must be mapped (condition **all** or **all+extents** in Table 17) and $W_{A\text{-}FA}$ must be 0.

The current default scoring parameters for RDR are given in Table 19.

Table 19 Default parameters for scoring RDR performance

***Element_Value* parameters**				
Attribute	Type	Subtype	Modality	Tense
AttrValue	1.00 for all types	n/a	1.00 for all modalities	n/a
$W_{err\text{-}attribute}$	1.00	0.70	0.75	1.00
***Relation mapping requirements* (**Table 17**) = "arguments"**				
W_{FA} = 0.75				
***Arguments_Value* parameters**				
"mapped" *arguments optional requirement* NOT *invoked* (Table 18**)**				
Both Arg-1 and Arg-2 arguments must be mappable (i.e., must have non-null *MAV*'s)				
"arithmetic" *averaging of argument scores*				
$W_{err\text{-}role}$ = 0.75		$W_{A\text{-}FA}$ = 0.00		

[20] Arg-1 and Arg-2 are the only roles for which the number of arguments is limited.

[21] If Arg-1/Arg-2 are reversed, *Element_Value* is penalized as if both type and subtype were incorrect, regardless of their actual values.

VDR SCORING

The VDR value score for a system is defined to be the sum of the values of all of the system's output event tokens, normalized by the sum of the values of all reference event tokens. The maximum possible VDR value score is 100 percent.

$$VDR\_Value_{sys} = \sum_i value\_of\_sys\_token_i \Big/ \sum_j value\_of\_ref\_token_j$$

The value of each system token is based on its attributes and arguments and on how well they match those of a corresponding reference token. A globally optimum correspondence between system and reference tokens which maximizes *VDR_Value* is determined and used, subject to the constraint of one-to-one mapping between system and reference tokens. System tokens and reference tokens are permitted to correspond only if they have some nominal basis for correspondence. The required nominal basis is selectable from the set of minimal conditions listed in Table 20. Note that the condition selected applies to both VDR and RDR.

Table 20 Conditions required for correspondence between system and reference event tokens

Condition	Description
arguments	At least one argument in the system token must be mappable to an argument in the reference token.
extents	The system and reference tokens must each have at least one mention extent in correspondence with the other.
both	Both the **arguments** condition and the **extents** condition must be met.
either	Either the **arguments** condition or the **extents** condition must be met.
all	All arguments in the reference token must be one-to-one mappable to arguments in the system token.
all+extents	Both the **all** condition and the **extents** condition must be met.

The value of a system token is defined as the product of two factors that represent both the inherent value of the token and how accurately the token's attributes and arguments are recognized.

$$Value(token) = Element\_Value(token) \cdot Arguments\_Value(token)$$

Element_Value is a function of the attributes of the system token and, if mapped, how well they match those of the corresponding reference token. The inherent value of a token is defined as the product of the token's attribute value parameters, *AttrValue*, for the attributes ***type*** and ***modality***. This inherent value is reduced for any attribute errors (i.e., for any difference between the values of system and reference attributes), using the error weighting parameters, $\{W_{err\text{-}attribute}\}$. If a system token is unmapped, then the value of that token is weighted by a false alarm penalty, W_{FA}.

$$Element\_Value(sys) = \begin{cases} \min\left(\prod_{\substack{attribute= \\ type,modality}} AttrValue(attribute_{sys}), \prod_{\substack{attribute= \\ type,modality}} AttrValue(attribute_{ref}) \right) \cdot \prod_{\substack{attribute= \\ type,subtype,modality, \\ genericity,polarity,tense}} W_{err-attribute} & \text{if mapped} \\ AttrValue(attribute_{sys}) \cdot W_{FA} & \text{if not mapped} \end{cases}$$

Arguments_Value is a function of the mutual argument value (*MAV*) between the arguments of the system token and, if mapped, those of the corresponding reference token.[22] An argument's *MAV* depends on the system argument's value (with respect to the putative reference argument) for each document in which the event is mentioned, $Value_{doc}(arg_{sys},arg_{ref})$, with this value being reduced for argument role errors (i.e., for a difference between the roles of system and reference arguments), $W_{err\text{-}role}$. Argument-level errors are accounted for using an incremental formulation of false alarm error. Specifically, loss of value at the argument level is viewed as a partial false alarm, and this loss of value is subtracted from the *MAV* after being weighted by a false alarm penalty factor, $W_{A\text{-}FA}$.

$$MAV_{doc}(arg_{sys}) = Value_{doc}(arg_{sys}, arg_{ref}) \cdot W_{err-role} - \left(Value_{doc}(arg_{sys}, arg_{sys}) - Value_{doc}(arg_{sys}, arg_{ref})\right) \cdot W_{A-FA}$$

[22] All arguments of a system token are considered to be unmapped for tokens that are themselves unmapped. Thus, for tokens that are unmapped, *Arguments_Value* will be negative. Note that *MAV* is negative for unmapped arguments, i.e., when $Value_{doc}(arg_{sys},arg_{ref}) = 0$.

If there is no corresponding reference argument for a system argument, then $Value_{doc}(arg_{sys},arg_{ref})$ is taken to be zero. There are several requirements that must be satisfied in order for a reference argument to be considered to be in correspondence to a system argument. These requirements for correspondence are listed in Table 21.

Table 21 Conditions required for correspondence between system and reference event arguments

Condition	Requirement
Always	The reference argument must be mappable to the system argument. That is, they must have at least one mention in correspondence.
If the "**mapped**" argument option is invoked	The reference argument must correspond to the system argument. That is, they must be mapped to each other at the argument level.

For each pairing of a system event token with a reference event token, an optimum correspondence between system arguments and reference arguments that maximizes *Arguments_Value* is determined and used. This optimum mapping is constrained to be a one-to-one mapping between system and reference arguments.

Arguments_Value is computed using one of two formulas, depending on whether the contribution of the various event arguments are averaged arithmetically or geometrically.

$$Arguments\_Value(sys) = \left\{ \begin{array}{l} \sum_{all\ arg_{sys}} \left(\sum_{\substack{all\ docs\ that \\ mention\ the\ relation}} MAV_{doc}(arg_{sys},arg_{ref}) \right) \quad \text{if arithmetic averaging} \\ \prod_{all\ arg_{sys}} \left(\sum_{\substack{all\ docs\ that \\ mention\ the\ relation}} MAV_{doc}(arg_{sys},arg_{ref}) \right) \quad \text{if geometric averaging} \end{array} \right\}$$

Note that geometric averaging is sensible only when the *MAV* value contributions exist and are positive for all reference arguments. Therefore, for geometric averaging, all reference arguments must be mapped (condition **all** or **all+extents** in Table 20) and $W_{A\text{-}FA}$ must be zero.

The current default scoring parameters for VDR are given in

Table 22 Default parameters for scoring VDR performance

***Element_Value* parameters**						
Attribute	Type	Subtype	Modality	Genericity	Polarity	Tense
AttrValue	1.00 for all types	n/a	1.00 for all modalities	n/a	n/a	n/a
$W_{err\text{-}attribute}$	0.50	0.90	0.75	1.00	1.00	1.00
***Event mapping requirements* (**Table 20**) = "arguments"**						
W_{FA} = 0.75						
***Arguments_Value* parameters**						
"mapped" *arguments optional requirement* NOT *invoked* (Table 21**)**						
"arithmetic" *averaging of argument scores*						
$W_{err\text{-}role}$ = 0.75			$W_{A\text{-}FA}$ = 0.50			

Event Nugget Annotation: Processes and Issues

Teruko Mitamura, Yukari Yamakawa, Susan Holm
Language Technologies Institute, Carnegie Mellon University, Pittsburgh PA
teruko@cs.cmu.edu, {yukariy, sh4s}@andrew.cmu.edu

Zhiyi Song, Ann Bies, Seth Kulick, Stephanie Strassel
Linguistic Data Consortium, University of Pennsylvania, Philadelphia PA
{zhiyi, bies, skulick, strassel}@ldc.upenn.edu

Abstract

This paper describes the processes and issues of annotating event nuggets based on *DEFT ERE Annotation Guidelines v1.3* and *TAC KBP Event Detection Annotation Guidelines 1.7*. Using Brat Rapid Annotation Tool (brat), newswire and discussion forum documents were annotated. One of the challenges arising from human annotation of documents is annotators' disagreement about the way of tagging events. We propose using Event Nuggets to help meet the definitions of the specific type/subtypes which are part of this project. We present case studies of several examples of event annotation issues, including discontinuous multi-word events representing single events. Annotation statistics and consistency analysis is provided to characterize the inter-annotator agreement, considering single term events and multi-word events which are both continuous and discontinuous. Consistency analysis is conducted using a scorer to compare first pass annotated files against adjudicated files.

1 Introduction

Annotating event mentions is useful for event detection tasks. It also is useful for detecting event coreference, subevent relations, event arguments, and realis values in corpora. This paper describes the processes and issues of annotating event nuggets based on the *DEFT ERE Annotation Guidelines v1.3* (LDC, 2014) (henceforth referred to as *Light ERE Guidelines*) and the *TAC KBP Event Detection Annotation Guidelines v1.7* (LTI, 2014) (henceforth referred to as *TAC KBP Event Guidelines*). Using the Brat Rapid Annotation Tool (brat)[1], we annotated files in newswire and discussion forums genres to create the corpus that supports the TAC KBP pilot evaluation for Event Nugget Detection as part of the DEFT program.

In this paper, we introduce the notion of event nugget and how event nuggets are annotated in the corpus. We discuss the issues that arose in the process of developing *TAC KBP Event Guidelines*, because they are important challenges for manual annotation and impact the quality of annotation for gold standard creation. Two major issues are (1) determining if an event meets the event type/subtype definitions and (2) deciding which words should be tagged within the span of a multi-word event nugget that represents a single event. We provide screen images of our annotation tool in order to give a complete picture of the annotation process. Finally, we present statistics to explain the characteristics of the corpus, such as the size of the corpus and the distribution of event type/subtypes. We discuss consistency analysis of inter-annotator agreement in terms of single word, multi-word continuous, and multi-word discontinuous event nuggets.

2 What is an Event Nugget?

It is challenging to provide clear-cut definitions of events, because many researchers define events differently. For example, in the Light ERE annotations, as well as in ACE*, Automatic Content Extraction) English Annotation Guidelines for Events* (LDC, 2005), an event is defined as an explicit occurrence involving participants. An event is something that happens at a particular place and time, and it can frequently be described as a change of state. The *Light ERE Guidelines* expect annotators to tag an event trigger, which is the smallest extent of text that expresses the occurrence of an event. Both ACE and Light ERE, only examples of a particular set of types/subtypes are tagged. An event trigger is usually a word or phrase. In many cases, event triggers are main verbs in sentences that in-

[1] Brat Rapid Annotation Tool (brat) was developed by Pontus Stenetorp et al. (2014). It is a web-based annotation tool.

Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT 2015, pages 66–76,
Denver, Colorado, June 4, 2015.

dicate the occurrence of the events. Annotating a main verb is relatively easy and is likely to produce a higher rate of inter-annotator agreement, because it allows annotators to pay more attention to a syntactic attribute of an event as well as its semantic feature. However, event triggers are not just verbs. Some nouns and adjectives can also express events (See examples in Section 3.1.).

In this study, we took a different approach to event annotations so that we would be able to annotate more complex events, which consist of multiple words taggable as events. For this reason, we decided to take a semantically oriented approach for annotation. New annotation guidelines were produced (*TAC KBP Event Guidelines*), based on the *Light ERE Guidelines* and *ACE*. To clarify the tagging of multiword events, we propose the idea of "event nugget," which is comprised of a semantically meaningful unit that expresses the event in a sentence. An event nugget can be either a single word (main verb, noun, adjective, adverb) or a continuous or discontinuous multi-word phrase.

The main reason why we propose event nugget annotation is to identify events accurately enough to meet the definitions of event types/subtypes in the *Light ERE Guidelines*. The type/subtype definitions restrict annotation to very specific types of events. Figuring out which events fall within the type/subtype definitions is a key issue to annotation. In the process of annotation, we have encountered cases in which multiple words could equally be considered as an event trigger. In many cases the multiple words are hard to separate from one another in terms of meaning (e.g., "hold a meeting", "serve a sentence", "send email"). Thus, we decided to annotate the maximum extent of text which meets the definition of the event types/subtypes provided by the *Light ERE Guidelines*. This approach allows annotators to tag all possible words that meet the definition of the event types/subtypes.

In addition to the annotation of the maximum extent of events, discontinuous tagging is another characteristic of event nugget annotation. (In order to clarify which words are in the same event nugget in this paper, we underline from the first word in a discontinuous multiword event nugget to the last word in the nugget. A dotted underline appears under words that are not part of the event nugget.) Discontinuous tagging allows annotators to tag words that do not lie next to each other but still belong to a multiword event nugget because they are all required to meet the definition, such as "The company **laid** 10 workers **off**," and "His **death sentence** was **carried out**."

Discontinuous tagging is very effective because it can be used to prevent violations of rules for annotation. For example, *TAC KBP Event Guidelines* as well as *Light ERE Guidelines* mention that non-main verbs should not be tagged. In sentences such as "His death sentence was carried out," annotators may want to tag "death sentence was carried out" to meet the definition of Justice_Execute events, since carrying out a death sentence means executing someone. However, tagging "was" violates the rule that non-main verbs are not taggable. In this case, tagging "death sentence" and "carried out" together as a discontinuous multiword event nugget not only meets the definition of Justice_Execute events but also does not violate the rule that "be" verbs should not be tagged.

The merits of event nugget annotation are summarized as follows: identification of events in a more semantically meaningful way and flexible annotation without violating annotation rules. In the next section, we present examples of event nuggets, using the following format to indicate the annotation: [Event Type_Subtype, REALIS]. Realis will be discussed in Section 3.3.

3 Types of Event Nuggets and REALIS

3.1 Single-Word Event Nuggets

As in ACE and Light ERE annotation, single-word event nuggets meet the definitions of event triggers for particular types/subtypes. Slightly modified in *TAC KBP Event Guidelines*, single-word event nuggets refer to words that meet the definitions of event types/subtypes by themselves. They are verbs (usually main verbs), nouns, adjectives, or adverbs. Below are some examples of single-word event nuggets. The words in **bold face** are event nuggets.

- The **attack** by insurgents occurred on Saturday. [Conflict_Attack, ACTUAL]
- Hillary Clinton was not **elected** president in 2008. [Personnel_Elect, OTHER]

There are some cases where multiple single-word event nuggets appear in the same sentence.

- Kennedy was **shot dead** by Oswald. [Conflict_Attack, ACTUAL], [Life_Die, ACTUAL]
- Three years ago, investors **bought** two stagnant web-hosting companies and **merged** them into what is now known as The Planet. [Transaction_Transfer-Ownership, ACTUAL], [Business_Merge-Org, ACTUAL]

Pronouns and other anaphors are also considered as single-word event nuggets if they refer to previous event mentions that meet the definitions of event types/subtypes.

- The **talks** between the Koreas were largely unsuccessful. **They** ended without agreement on Monday. [Contact_Communicate, ACTUAL], [Contact_Communicate, ACTUAL]

3.2 Complex (Multi-Word) Event Nuggets

Complex event nuggets are multi-word phrases (or compounds) that construct semantic units that meet the definitions of event types/subtypes. Those units can be continuous or discontinuous. Multi-word event nuggets take various forms such as verb+noun, verb+particle/adverb, noun+noun, and so on. The words underlined and in **bold face** are multi-word event nuggets that represent a single event.

- Foo Company had **filed Chapter 11** in 2000. [Business_Declare-Bankruptcy, ACTUAL]
- The police investigated the **murder incident**. [Conflict_Attack, ACTUAL]

Discontinuous tagging is one of the characteristics of annotation of multi-word event nuggets. This type of tagging is useful because it captures event nuggets accurately without missing important components of meaning. Below are the examples of discontinuous tagging of multi-word event nuggets.

- The court **found** him **guilty**. [Justice_Convict, ACTUAL]
- His **death sentence** was **carried out**. [Justice_Execute, ACTUAL]
- All **charges** were **dropped** against him last year. [Justice_Acquit, ACTUAL]

Multi-word event nuggets that represent single events are tagged either continuously or discontinuously depending on the particular construction of the semantic units that meet the definitions of the event types/subtypes in each sentence.

For example, consider the definition of Justice_Sue: "A SUE event occurs whenever a court proceeding has been initiated for the purposes of determining the liability of a PERSON, ORGANIZATION or GPE accused of committing a crime or neglecting a commitment." The three examples below illustrate event nuggets for Justice_Sue events. (For clarification, strikethrough denotes an event that is not part of the event nugget being illustrated.)

- His lawyer should **file** a **lawsuit**. [Justice_Sue, OTHER]
- His lawyer should **sue**. [Justice_Sue, OTHER]
- His lawyer should ~~contest~~ the **lawsuit**. [Justice_Sue, OTHER]

The noun+verb combination of "file" and "lawsuit" meet the definition of Justice_Sue as a court proceeding having been initiated. A lawsuit is a court proceeding, and filing refers to its initiation, which is a part of the court proceeding. The two words in combination express the "doing" of the SUE event and meet the definition of Justice_Sue. The single verb "sue" can also be used to meet this definition, as can the single noun "lawsuit". However in the third sentence, "contest" is separate from the lawsuit event and does not belong to the event nugget. To contest a lawsuit is an action of the defense team in response to an existing lawsuit. There is currently no Justice Subtype defined in the *Light ERE Guidelines* to fit this contest event.

3.3 REALIS

In our annotation, event nuggets are annotated with three types of REALIS: ACTUAL, GENERIC, and OTHER. REALIS relates to whether or not an event occurred (LTI, 2014).

The REALIS of ACTUAL is used when the event actually happened at a particular place and time, involving specific entities. Both ongoing events and events that have ended are tagged ACTUAL. For example, "He **emailed** her about their plans [Contact_Communicate, ACTUAL]."

The REALIS of GENERIC is used for events that refer to general events involving types or categories of entities. GENERIC is also used for taggable event nuggets which appear in statistics or demographic information. For example, "People **die** [Life_Die, GENERIC]."

The REALIS of OTHER will be used for events that are neither ACTUAL nor GENERIC. If it is determined that an event meets the definition of a type/subtype and it is not an ACTUAL or GENERIC event, it can simply be tagged OTHER. For example, "He plans to **meet** with both political parties [Contact_Meet, OTHER]."

In the case of GENERIC events which also qualify as OTHER (e.g., negated generic) or ACTUAL (e.g., past generic, habitual generic), GENERIC is used, not OTHER or ACTUAL.

4 Event Types/Subtypes

The *TAC KBP Event Guidelines* and the *Light ERE Guidelines* share the same 33 event types/subtypes in particular areas, such as Life, Movement, Business, Conflict, Personnel, Transaction, and Justice, which were originated in the *ACE Guidelines* (LDC, 2005).

The complete set of event types/subtypes is: Life (Be-Born, Marry, Divorce, Injure, Die), Movement (Transport-Person), Business (Start-Org, End-Org, Declare-Bankruptcy, Merge-Org), Conflict (Demonstrate, Attack), Contact (Meet, Communicate), Personnel (Start-Position, End-Position, Nominate, Elect), Transaction (Transfer-Ownership, Transfer-Money), Justice (Arrest-Jail, Release-Parole, Trial-Hearing, Charge-Indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon).

- John Doe was **born** in Casper, WY. [Life_Be-Born, ACTUAL]
- Roosevelt and his family immediately **departed** for Buffalo. [Movement_Transport-Person, ACTUAL]
- A car bomb **exploded** in central Baghdad. [Conflict_Attack, ACTUAL]

5 Annotation Challenges

One of the main challenges in the development of annotation guidelines is that there is always some disagreement about what should (not) be taggable. In this section, we present some examples of disagreements, which we experienced in the process of developing annotation guidelines, as case studies.

The first case is related to annotating implied events which are contained within nouns referring to persons (e.g., "protestor", "assailant", "killer"). The second case concerns prepositional phrases (e.g., "in prison", "behind bars"), which seem to meet the definitions of event types/subtypes. The third case involves annotating nouns that refer to the consequences or results of events (e.g., "injury", "body", "funeral"), which could be considered as either an entity or an event by individual annotators. The fourth case occurs when only a portion of a word indicates an event (e.g., "antiwar", "post-war", "ex-husband", "ex-wife"). The last case is discontinuous tagging of event nuggets. Although discontinuous tagging is effective for capturing the semantically meaningful unit of event nuggets, the consistency (See Table 5) of discontinues event nuggets is not as good as singe token event nugget.

In the case studies below, the words in ***italic bold*** are controversial or in issue.

Case Study 1: Is a person an event?

- Two other ***assailants*** have committed suicide.
- Here is the KICKER: As reported by local news stations DOZENs of ***protestors*** showed up to protest.
- On the grounds of legality, according to the Geneva Convention, members of regular armed forces – involved in conflicts – are the only persons who may be considered lawful ***combatants*** and authorized to use lethal force.

The words such as "assailants", "protesters", and "combatants" imply the occurrence of events, as we can see by paraphrasing them as "a person who assailed (assaulted) someone," "people who are protesting," and "people who combat." If annotators take the implied occurrences into consideration, those words will be tagged as event nuggets. However, those words actually refer to the "people" themselves. People are not events. Tagging them as events means that we tag implied events. In a similar fashion, some annotators may be tempted to tag "the **dead**" as an event nugget, but others do not because they think that "the dead" refers to dead people. It is critical for annotators to consider the implications of implied events when they tag. If implied events are to be tagged, rules should be explicitly stated to guide annotators as to which implied events should be tagged, and which implied events should not be tagged.

Case Study 2: Is a prepositional phrase taggable?

- A former militant of the French far-left group Action Directe, Georges Cipriani, left prison

on parole on Wednesday after 23 years ***behind bars*** for two high-profile murders.

- Prosecutors have said Chen could face life ***in prison*** if convicted on all counts, including embezzlement and bribe-taking."

The phrases "behind bars" and "in prison" indicate that the agent was (or would be) imprisoned and could be tagged as Justice_Arrest-Jail events. They are, however, prepositional phrases that describe a certain state (i.e. the state of physically residing in a particular place). There is some debate whether or not states are taggable as events. Especially in the case of prepositional phrases, it is difficult for annotators to decide whether those phrases should be tagged, since they could be considered to refer to states and sound less eventive.

Case Study 3: Is it an event or the result of an event?

- Why was Trayvon's ***body*** laying 12 hours in the Morgue?
- A cry for the men to be hanged went up almost immediately after the woman died of her ***injuries***, …
- And those already existing time place and manner restrictions were utilized at Matthew Snyder's ***funeral***, with the result that the family never even knew WBC was there.

The words in ***italic bold*** indicate the consequence or result of certain events. For example, the type of "body" referred to in the first example only exists after a Life_Die event has occurred. "Injuries" exist on or in a person's body after (s)he has experienced a Life_Injure or Conflict_Attack event. A "funeral" is a ceremony that occurs after a Life_Die event has happened. Since "body", "injuries" and "funeral" are words that are closely related to taggable event types/subtypes, annotators may be tempted to tag those words as event nuggets. However, it is necessary to differentiate the consequence/result of an event from an event itself.

Case Study 4: Is a portion of a word taggable?

- U.N. Secretary General Kofi Annan said this week that the body has no interest in policing a ***postwar*** Iraq, …
- We were so proud of forming an ***antiwar*** bloc with France and Germany …
- Jurassic Park creator Crichton agrees to pay ***ex-wife*** 31 million dollars

The decision on whether a portion of a word should be tagged also causes disagreement among annotators. Some annotators may think it not appropriate to break a word into chunks, or others may tag a part of a word only if it is hyphenated. This case study raises the issue on how events are defined in relation to word level structure. Semantically, both "war" and "ex" meet the definitions of event types/subtypes. However, it is unclear whether the entire word ("postwar", "antiwar", "ex-wife") should be tagged. Is "antiwar" a Conflict_Attack event, for instance? It is necessary to have a clear rule for this type of tagging.

Case Study 5: Tagging Discontinuous Multiword Event Nuggets

In our corpus with 3,798 event nuggets, there were 209 discontinuous nuggets, a ratio of 5.5%. The discontinuous event nuggets appear in various forms such as verb+noun, verb+particle/adverb, verb+adjective, and verb+prepositional phrase. Among those patterns, the most frequent one is a verb+noun compound (83%), where a noun is the direct object of the verb. This pattern appears in a passive form as well.

- today I **got** a **letter** from the hospital [Contact_Communicate, ACTUAL]
- where was the father when the **shot** was **fired** not more than a 1000 feet away? [Conflict_Attack, ACTUAL]

These discontinuous events are tagged because multiple words in the sentence are important semantic components of their event type/subtype definitions. For example, the word, "get" is used to create various event types such as "get money" (Transaction_Transfer-Money) and "get a job" (Personnel_Start-Position). Thus, tagging a verb and a noun together as one event seems important to differentiate a particular event type from the others. In the second example, both "shot" and "fired" are taggable as events and it is hard to ignore either of them as not taggable due to the close relationship between the "doing" of an event and event itself. A verb+noun compound appears very often in the following event types/subtypes: Transaction_Transfer-Money (23%), Contact_Communicate (18%), and Conflict_Attack (10%).

Part of speech patterns for discontinuous tagging include verb+particle/adverb, which is 14% of the entire discontinuous tagging. This form appears most often in Movement_Transport-Person (68%).

- …**took** us **in** for a interview…[Movement_Transport-Person, ACTUAL]
- ... i **put** the thread **up** because i really did want some opinions…[Contact_Communicate, ACTUAL]

Some annotators may only tag main verbs because they think adverbs and particles are modifying the verbs, but others may tag verb+adverb/particles together because they feel that the adverb/particles signify a different meaning from just the verbs alone. As shown Table 5, it is not as easy to consistently annotate multi-word event nuggets as it is to consistently annotate single-word event nuggets. However, the percentage of multi-word event nuggets is so low that it may not significantly affect overall event nugget detection performance.

We continue to work on reaching agreement on the optimal method of handling of these four types of controversial event nuggets in order to better represent the deeper semantics of texts. The very low frequency of discontinuous event nuggets does not mean that they should be ignored to achieve higher inter-annotator agreement. Clear rules for these cases should be laid out for future tasks on event nugget detection.

6 Brat Rapid Annotation Tool (brat)

Our annotation was conducted using Brat Rapid Annotation Tool (brat). This tool allows for customization of tags, such as event types/subtypes, realis types, types of entities/arguments, types of event links, and provides a means to add notes for questionable mentions. In addition, brat supports discontinuous tagging and side-by-side comparison of two files.

The actual procedure of annotation and the review of applied tags are relatively simple with this user-friendly application. Clicking on a word to be tagged opens a window where annotators can select tags, such as event types/subtypes and realis. After a word has been tagged, when the cursor is moved over the tag, a small box appears, displaying the assigned event type and realis for review. Screenshots of brat are shown in the Appendix.

7 Data Selection and Preparation

We produced training and evaluation (eval) data to support the Event Nugget evaluation as a pilot TAC KBP evaluation. The data includes both formal newswire text (NW) and informal discussion forums (DF), drawn from a pool of data also labeled for the DARPA DEFT Program's Light Entities, Relations and Events (Light ERE) task (Song et al., 2015), and/or the NIST TAC KBP Evaluation Event Argument Task (Ellis et al., 2014), with the goal of ultimately being able to take advantage of multiple styles of event annotation on the same data. Documents for the current task were carefully selected from this pool to optimize coverage of as many of the event types and subtypes as possible, with a goal of at least five instances of each type-subtype combination. The training data consists of 151 documents, while the eval data contains 200 documents. Table 1 shows the genre distribution as well as token counts for each partition.

Partition	Training		Eval	
Genre	NW	DF	NW	DF
Documents	77	74	101	99
Tokens	44,962	70,427	50,997	169,740

Table 1. Event Nugget Data Profile

While the Light ERE and KBP Event Argument tasks rely on character offsets for annotation and scoring, the Event Nugget Tuple Scorer[2] (Liu, Mitamura & Hovy, 2015) requires tokenized data. Therefore, prior to annotation, all selected documents were automatically tokenized in the Penn English Treebank style. No manual correction was performed on the tokenization due to time constraints.

8 Corpus and Consistency Analysis

8.1 Corpus

Experience with event annotation for Light ERE and ACE (Doddington et al., 2004) and related tasks suggests that a major challenge for annotation consistency is poor recall – human annotators are not highly consistent in recognizing that a mention has occurred. To reduce the impact of this known issue for the Event Nugget task, two anno-

[2] Event Nugget Tuple refers to the tuple made up of the nugget, event type/subtype, and realis.

tators independently labeled each document (two first pass annotation passes, referred to as FP1 and FP2 below); a senior annotator then adjudicated discrepancies to create a gold standard. The team consisted of four first pass annotators, two of whom were also adjudicators. The effort was made to ensure that annotators did not adjudicate their own first pass files, but due to time constraints and the pilot nature of the task, in some cases there was overlap.

The gold standard training data has 3,798 event nuggets annotated in total, while the eval data has 6,921 event nuggets. Table 2 shows the distribution of event nuggets by genre and realis type for each partition.

Realis Attribute	Training		Eval	
	NW	DF	NW	DF
Generic	202	383	245	981
Other	346	406	448	1271
Actual	1313	1132	1752	2224
Total	3798[3]		6921	

Table 2. Realis Annotation of Event Nuggets

Figure 1 (in Appendix) shows the distribution of each type-subtype combination in the training and eval data. Conflict_Attack has the highest representation in both training (579) and eval (791). Justice_Extradite has the lowest count in training data (3), while Life_Be-Born is least frequent in the eval data (19). Despite our efforts to manually select documents to maximize coverage for all type-subtype combinations, the corpus does not include any occurrences of Business_End-Org or Personnel_End-Position.

8.2 Consistency Analysis

We examined annotation consistency and quality by comparing different passes of the eval set annotation using the Event Nugget Tuple Scorer (Liu, Mitamura, & Hovy, 2015) developed for the event nugget evaluation task. This scorer treats one file as "gold" and the other as "system", and matches each nugget in the gold file to one or more nuggets in the system file. This mapping is based on the overlap of the nugget spans. By nugget span, we mean the exact list of tokens, continuous or discontinuous, that make up an event nugget. However, each system nugget can only be mapped to one gold nugget. For each gold nugget, the scorer computes type and realis accuracy scores based on the values for the gold nugget and all the system nuggets that are mapped to it.

The scorer produces three scores for each file. The first is an F-measure for the nugget spans, based on the mapping from gold to system nuggets, as well as ''false alarms" in the system file that are not mapped to any nuggets in the gold file. The type and realis scores for each gold mention are also cumulatively summed up, producing a type and realis score for the file. The type and realis scores are therefore tied to the F-measure score of the nugget spans. We used this scorer rather than the ACE (NIST, 2005) scorer since this scorer was designed for the event nugget evaluation task, and so seemed the most appropriate to use for evaluation of annotation consistency and quality of this corpus.

We examined annotation consistency by comparing the two independent first passes of annotation (FP1 and FP2), with the results shown in the column FP1 vs. FP2 in Table 3. We also evaluated improvement in annotation quality in the workflow by comparing the adjudicated (ADJ) and first (FP1 and FP2) passes, shown in the columns ADJ vs. FP1 and FP2 in Table 3. The noticeable improvement in score shows the advantage of including adjudication as part of the annotation process. (For IAA purposes, there is obviously no gold or system, but in order to use the scorer we arbitrarily treated one file as the "gold".)

	FP1 vs. FP2	ADJ vs. FP1	ADJ vs. FP2
Span	69.0	78.2	89.3
Type	68.2	71.7	84.3
Realis	60.0	63.2	85.7

Table 3. Scores for Event Nugget Eval Set Annotation

To gain some further insight into these numbers we expanded the analysis in two directions. First, we compared the FP1 vs. FP2 event nugget consistency with the FP1 vs. FP2 annotation consistency on the ACE 2005 training data (Walker et al., 2006). There is also a scorer that was developed for ACE (NIST, 2005), but we used the Event Nugget Tuple evaluation scorer so that we could score both sets of data for this comparison as in the

[3] 16 event nuggets in the training set did not receive a realis attribute, due to annotation error.

event nugget evaluation. This necessitated converting the ACE files into the format for event nuggets used for the current scorer. We used the ‘‘anchor’’ string of the ACE event mention as the nugget span, the ‘‘type’’ and ‘‘subtype’’ of the ACE event mention as the nugget type, and the ‘‘modality’’ of the ACE event mention as the nugget realis value. The results are shown in Table 4. The ACE FP1 vs. FP2 scores in Table 4 are somewhat lower than the FP1 vs. FP2 scores for the event nugget annotated data. However, while we have converted the format and used the same scorer, the annotation task is not identical, so this can only be taken as a rough comparison. There is greater difference between the ADJ vs. FP1, FP2 scores for the event nugget data than the ACE data. The event nugget task had a smaller annotation team than for ACE, and it is likely that more of the adjudication annotators for event nugget annotation also did the FP2 pass than was the case for ACE.

	FP1 v. FP2	ADJ v.FP1	ADJ v. FP2
Span	64.8	79.3	81.8
Type	62.2	70.4	75.6
Realis	56.1	68.0	73.0

Table 4. Scores for ACE 2005 Training Annotation

Second, we wished to determine also if there was a difference in the annotation consistency and quality of event nugget spans depending on whether the span consists of only one token as compared to those that are multiple tokens, either continuous or discontinuous. We decomposed the span F-measure in Tables 3 and 4 based on these criteria. We did this by modifying the event nugget scoring program to optionally ignore nuggets depending on their span. For example, when we wished to compare annotations for which the span is a single token, we simply ignored all nuggets with spans of more than one token. Likewise, when comparing nuggets for which the span consists of discontinuous multiple tokens, all nuggets for which the span was either a single token or multiple continuous tokens were ignored.

We ran this modified scorer in different modes to use (1) all nuggets (as before), (2) only nuggets that consist of a single token, ignoring all others, (3) only nuggets that consist of multiple continuous tokens, (4) only nuggets that consist of multiple discontinuous tokens, and (5) only nuggets that consist of multiple tokens, whether continuous or not. Mode (1) is the same as the score reported for the spans in Tables 3 and 4, and modes (2)-(5) in effect break this down into subcomponents. The results are shown in Table 5. ACE annotation did not allow discontinuous multiple token mentions, and so there are no results listed for ACE for (4) and (5).

The results for the consistency agreement between FP1 and FP2 show a similar fall in score for both the event nugget data and the ACE 2005 training data, when considering only multiple continuous tokens. The score climbs back up a little for the event nugget FP1 vs. FP2 score when considering (5) either continuous or discontinuous multiple tokens, as compared with either (3) only multiple continuous or (4) only multiple discontinuous. The reason for this is that there are cases where one file has an event nugget with a continuous multiple token span such as “got jail time” while the other has the corresponding event nugget with a multiple discontinuous span such as ‘‘got time’’. In (3) or (4), only one or the other would be included in the comparison, whereas in (5) and (1) both would be included, allowing for partial match instead of a miss. Similarly, there are cases where one file has a single token span for a nugget while the other file has a multiple token span for the corresponding nugget, and so it is only in (1) that both would be included, allowing for a partial match instead of a miss.

These more fine-grained nugget span scores for FP1 vs. FP2 show that single-token nuggets are annotated more consistently than multi-token nuggets. Considering just the multi-token nuggets, there is little difference in consistency of annotation between continuous and discontinuous spans. The ADJ vs. FP1 / ADJ vs. FP2 results show that including adjudication annotation lessens any difference in annotation quality for nuggets depending on whether the span is single or multi-token.

In future work on this consistency analysis, we will also go in the other direction, and convert the event nugget data into the ACE format so that it can be evaluated using the ACE scorer (NIST, 2005), ensuring that the comparison of inter-annotator consistency is not overly affected by details of particular scoring algorithms.

	Event Nugget				ACE 2005 Training			
	FP1 vs. FP2		ADJ vs. FP1 / ADJ vs FP2*		FP1 vs. FP2		ADJ vs. FP1 / ADJ vs. FP2	
	Span F-meas	Ratio**	Span F-meas	Ratio	Span F-meas	Ratio	Span F-meas	Ratio
(1) All mentions	69.0	100%	78.2/89.3	100%	64.9	100%	79.3/81.8	100%
(2) Single-token	67.7	90.0%	77.0/88.9	87.7%	65.0	94.6%	79.2/81.6	95.2%
(3) Multiple cont.	45.3	6.1%	57.7/84.4	6.8%	44.2	5.4%	70.8/70.6	4.8%
(4) Multiple discont.	43.0	4.0%	57.5/84.1	5.5%	NA	NA	NA	NA
(5) Multiple all	46.0	10.1%	59.0/85.4	12.3%	NA	NA	NA	NA

Table 5: Decomposing the Span Scores for Nugget and Trigger Span

* The two figures represent ADJ compared to FP1 (before the slash) and ADJ compared to FP2 (after the slash).

** Event nugget type per all event nuggets.

9 Conclusion

This paper first describes the processes of event nugget annotation using a brat tool and issues which arose in the process of developing *TAC KBP Event Guidelines*. We present complex cases that cause annotators' disagreement on tagging. Questions are raised about implied events, states vs. events, results of events, tagging portions of words, and discontinuous tagging. Second, the paper explains the creation of a tagged event nugget corpus and provides annotation statistics and consistency analysis comparing the first pass annotations, and also a comparison of adjudicated files with first pass files using the Event Nugget Tuple Scorer. The analysis shows that single-word nuggets are tagged more consistently than multi-word nuggets and that adjudication is very important for improving the quality of annotation.

Reconciliation of annotation disagreement is crucial in terms of not only the development of annotation guidelines but also the quality of annotation. This is closely associated with how an event nugget is defined and clarification of tagging rules. Resolving the issues surrounding event type/subtype definitions will be very helpful not only for future studies on event nugget detection but also studies on event coreference, subevent relations, and event arguments.

Acknowledgments

This material is based on research sponsored by Air Force Research Laboratory and Defense Advanced Research Projects Agency under agreement numbers FA8750-13-2-0045 and FA8750-12-2-0342. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory and Defense Advanced Research Projects Agency or the U.S. Government.

We are thankful to Jun Araki, Zhengzhong "Hector" Liu, and Volkan Cirik for their assistance in the annotation tool and data calculation.

References

George Doddington, Alexis Mitchell, Mark Przbocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The Automatic Content Extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, May 24-30.

Joe Ellis, Jeremy Getman, and Stephanie M. Strassel. 2014. Overview of Linguistic Resources for the TAC KBP 2014 Evaluations: Planning, Execution, and Results. In *Proceedings of TAC KBP 2014 Workshop*, National Institute of Standards and Technology, Gaithersburg, Maryland, USA, November 17-18, 2014.

Language Technologies Institute. 2014. *TAC KBP Event Detection Annotation Guidelines, Version 1.7*, Language Technologies Institute, CMU, September 12, 2014.

Linguistic Data Consortium. 2014. *DEFT ERE Annotation Guidelines: Events Version 1.3*, March 13, 2014.

Zhengzhong Liu, Teruko Mitamura, Eduard Hovy. 2015. "Evaluation Algorithms for Event Nugget Detection: A Pilot Study". To appear in the Proceedings of the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation. NAACL-HLT 2015.Linguistic Data Consortium. 2005. *ACE (Automatic Content Extraction) English Annotation Guidelines for Events, Version 5.4.1 2005.05.09*.

National Institute of Standards and Technology. 2005. *The ACE 2005 Evaluation Plan*. http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/ace05-evalplan.v3.pdf

Zhiyi Song, Ann Bies, Tom Riese, Justin Mott, Jonathan Wright, Seth Kulick, Neville Ryant, Stephanie Strassel, Xiaoyi Ma. Submitted. From Light to Rich ERE: Annotation of Entities, Relations, and Events.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou and Jun'ichi Tsujii (2012). brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations Session at EACL 2012*.

Christopher Walker, Stephanie Strassel, Julie Medero, Kazuaki Maeda. 2006. *ACE 2005 Multilingual Training Corpus*. Linguistic Data Consortium Catalog No.: LDC2006T06.

Appendix

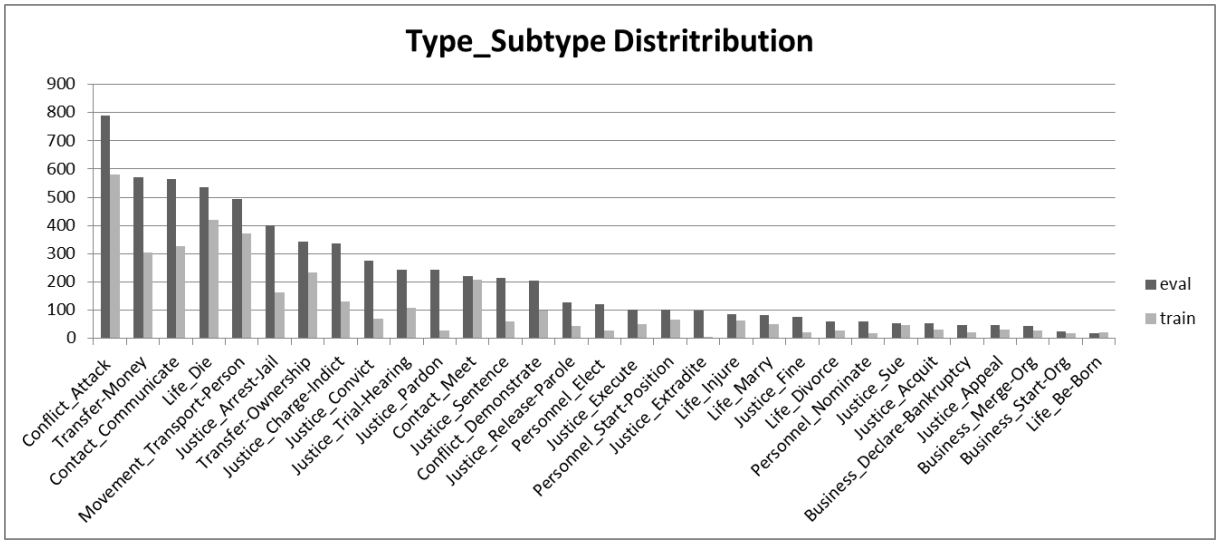
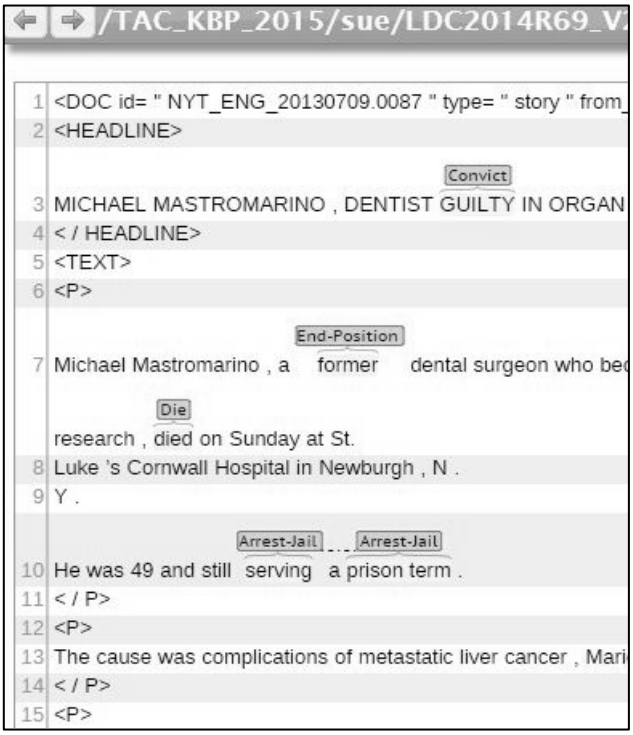
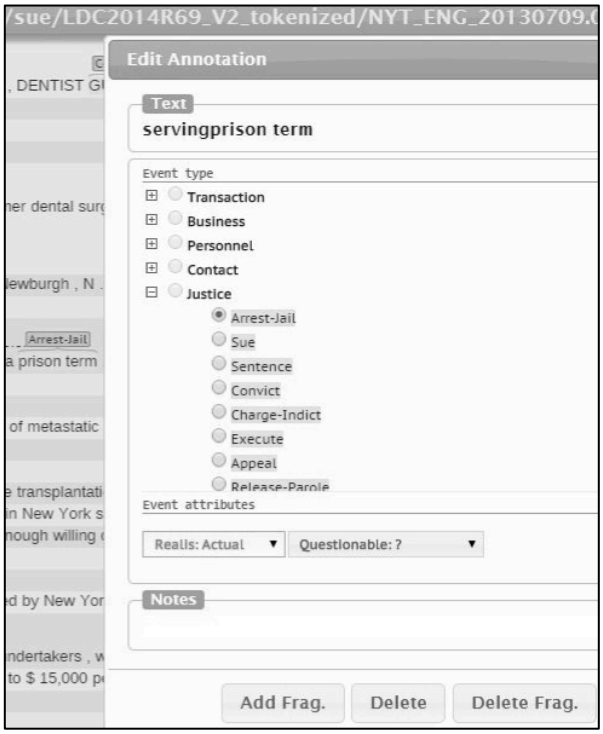


Figure 1. Type and Subtype Distribution in Event Nugget Annotation



Screenshot 1. Brat tool main annotation screen



Screenshot 2. Brat tool pop-up window

An Overview of Event Extraction from Text

Frederik Hogenboom, Flavius Frasincar, Uzay Kaymak, and Franciska de Jong

Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, the Netherlands

{fhogenboom, frasincar, kaymak, fdejong}@ese.eur.nl

Abstract. One common application of text mining is event extraction, which encompasses deducing specific knowledge concerning incidents referred to in texts. Event extraction can be applied to various types of written text, e.g., (online) news messages, blogs, and manuscripts. This literature survey reviews text mining techniques that are employed for various event extraction purposes. It provides general guidelines on how to choose a particular event extraction technique depending on the user, the available content, and the scenario of use.

1 Introduction

With the increasing amount of data and the exploding number of digital data sources, utilizing extracted information in decision making processes becomes increasingly urgent and difficult. An omnipresent problem is the fact that most data is initially unstructured, i.e., the data format loosely implies its meaning [9] and is described using natural, human-understandable language, which makes the data limited in the degree in which it is machine-interpretable. This problem thwarts the automation of for example vital information retrieval (IR) and information extraction (IE) processes – used for decision making – when involving large amounts of data.

Text Mining (TM) [15] is concerned with information learning from preprocessed text (e.g., containing identified parts of speech or stemmed words). By means of text mining, often using Natural Language Processing (NLP) [22] techniques, information is extracted from texts of various sources, such as news messages and blogs, and is represented and stored in a structured way, e.g., in databases. A specific type of knowledge that can be extracted from text by means of TM is an event, which can be represented as a complex combination of relations linked to a set of empirical observations from texts.

An example of an event is an acquisition. If we consider the representation `<Company> <Buy> <Company>`, words identified in text referring to companies are linked to the concept `<Company>`, and (conjugations of) verbs having the meaning of acquisition are associated with `<Buy>`. Representations of this event can be extracted from news message headers such as "*Google acquires Picnik*", "*Lala bought by Apple*", or "*Skype sold to Microsoft*".

Event extraction from unstructured data such as news messages could be beneficial for IE systems in various ways. For instance, being able to determine events could enhance the performance of personalized news systems [2, 10], as news messages can be selected more accurately, based on user preferences and identified topics (or events). Furthermore, events can be useful in risk analysis applications [3], monitoring systems [17], and decision making support tools [36].

Extracted events are also extensively applied within the medical domain [6, 38], where event parsers are utilized for extracting medical or biological events like molecular events from corpora. Another possible – but less researched – application of event extraction lies within the field of algorithmic trading, representing the use of computer programs for entering trade orders with algorithms deciding aspects like timing, price, and quantity of an order. Financial markets are extremely sensitive to breaking news [24]. Economic events like mergers and acquisitions [31], stock splits [14], dividend announcements [23], etc., play a crucial role in the daily decisions taken by brokers, where brokers can be humans or machines. Besides being able to process news faster, machines are able to deal with larger volumes of emerging news, having access to more information than we humans do, and thus making better informed decisions.

Given the promising potential for applications of event extraction, and assuming that the challenges of real-time extraction and combination of events can be tackled adequately, it is worthwhile to investigate which text mining techniques are appropriate for this purpose. The current body of literature is lacking a high-level survey on event detection in text. Therefore, the goal of this paper is to review existing approaches to event extraction from text. We aim for providing general guidelines on selecting the proper text mining techniques for specific event extraction tasks, taking into account the user and its context. For this, we strive for a similar overview of performance aspects and recommendations as has been developed for cross-lingual research systems [25]. The work presented herein is a first step, focussing specifically on event extraction from text. The recognition of the space and time event dimension in text is considered outside the scope of this paper.

Throughout this paper we evaluate event extraction approaches using several criteria. For this, we review data that are available in the literature and distinguish between the categories high, medium, and low. First of all, we investigate the amount of data needed for each approach. Moreover, the amount of required domain knowledge is evaluated, together with the required amount of user expertise. Finally, we also discuss the interpretability of the results.

This paper continues with an elaboration of approaches to event extraction in Section 2. Subsequently, Section 3 presents a discussion on the event extraction approaches introduced in this survey. Finally, Section 4 concludes the paper.

2 Event Extraction

We distinguish between three main approaches to event extraction, in analogy with the classic distinction that is made in the field of modeling. First, there

are data-driven approaches, described in Section 2.1, which aim to convert data to knowledge through the usage of statistics, machine learning, linear algebra, etc. Second, we distinguish expert knowledge-driven methods as discussed in Section 2.2, which extract knowledge through representation and exploitation of expert knowledge, usually by means of pattern-based approaches. Finally, the hybrid event extraction approaches elaborated on in Section 2.3 combine knowledge and data-driven methods.

2.1 Data-Driven Event Extraction

Data-driven approaches are commonly used for natural language processing applications. These approaches rely solely on quantitative methods to discover relations. Data-driven approaches require large text corpora in order to develop models that approximate linguistic phenomena. Furthermore, data-driven text mining is not restricted to basic statistical reasoning based on probability theory, but encompasses all quantitative approaches to automated language processing, such as probabilistic modeling, information theory, and linear algebra.

One could distinguish between many approaches, e.g., word frequency counting, ranking by means of the Term Frequency – Inverse Document Frequency metric, word sense disambiguation, n-grams, and clustering. Despite their differences, all approaches focus on discovering statistical relations, i.e., facts that are supported by statistical evidence. Examples of discovered facts are words or concepts that are (statistically) associated with one another. However, statistical relations do not necessarily imply semantically valid relations, nor relations that have proper semantic meaning.

Several examples of the usage of data-driven text mining approaches for event extraction can be found in literature. For instance, in their 2009 work, Okamoto et al. [27] elaborate on a framework for detection of occasional or local events, which employs hierarchical clustering techniques. While clustering itself could already yield promising results for event extraction, the authors of [21] make use of a combination of weighted undirected bipartite graphs and clustering in order to extract key entities and significant events from daily web news. Clustering techniques are also employed by Tanev et al. [34], who also aim for real-time news event extraction, but focus especially on violence and disaster events. The authors make use of automatic tagging of words and the presented framework is designed to automatically learn patterns from discovered events. Lastly, the authors of [19] also employ word-based statistical text mining in their work from 2005. The authors elaborate on a framework aimed at news event detection, based on support vector machines.

A drawback of the discussed data-driven methods to event extraction is that they do not deal with meaning explicitly, i.e., they discover relations in corpora without considering semantics. Another disadvantage of statistics-based text mining is that a large amount of data is required in order to get statistically significant results. However, since these approaches are not based on knowledge, neither linguistic resources, nor expert (domain) knowledge are required.

2.2 Knowledge-Driven Event Extraction

In contrast to data-driven methods, knowledge-driven text mining is often based on patterns that express rules representing expert knowledge. It is inherently based on linguistic and lexicographic knowledge, as well as existing human knowledge regarding the contents of the text that is to be processed. This alleviates problems with statistical methods regarding meaning of text. Information is mined from corpora by using predefined or discovered linguistic patterns, which can be either lexico-syntactic patterns [11, 12] or lexico-semantic patterns [2]. The former patterns combine lexical representations and syntactical information with regular expressions, whereas the latter patterns also make use of semantic information. Semantics are usually added by means of gazetteers, which use the linguistic meaning of text [7, 8], or by means of ontologies [10, 32].

Several attempts have been made for extracting events using pattern-based approaches to text mining. Both – mostly manually created – lexico-syntactic and lexico-semantic patterns are used; the former more often than the latter. For instance, in their 2009 work, Nishihara et al. [26] extract personal experiences from blogs by means of three keywords (place, object, and action) that together describe an event. For this, sentences are split using lexico-syntactic patterns. A similar approach can be found in [1], where the authors focus on pattern-based relation and event extraction. Here, lexico-syntactic patterns are employed in order to discover a wide range of relations and events in the domains of finance and politics. The authors of [38] elaborate on a methodology to extract events using a general-purpose parser and grammar applied to the biomedical domain. To this extent, lexico-syntactic patterns are employed that define the argumentation structures within texts. Hung et al. [13] elaborate on a framework that can be employed for mining the Web for event-based commonsense knowledge by using lexico-syntactic pattern matching and semantic role labeling. A large number of raw sentences that possibly contain target knowledge is collected through Web search engines. Web queries are formulated based on a set of lexico-syntactic patterns. After labeling the semantic roles, i.e., defining the relationships that syntactic arguments have with verbs, knowledge is extracted and stored in a database. A final example of the employment of lexico-syntactic patterns can be found in the work of Xu et al. [37]. Here, the authors envisage the usage of lexico-syntactic patterns in order to learn patterns from texts on prize award events, in the form of bootstrapping-oriented unsupervised machine learning, initialized with lexico-syntactic pattern seeds.

In pattern-based event extraction, concepts that have specific meanings and/or relationships are required, but either they are not available or they are not used due to the lack of pattern expressivity (i.e., in lexico-syntactic patterns). To solve this, lexico-semantic patterns are employed. These patterns are used for various purposes. In an attempt to discover event patterns from stock market bulletins, the authors of [20] analyze tagged corpora by means of gazetteering semantic concepts that are based on a (financial) domain. Cohen et al. [6] employ a concept recognizer on a biological domain in order to extract medical events from corpora, thus taking into account the semantics of domain concepts.

A similar approach is used by Vargas-Vera and Celjuska [35], who propose a framework for event recognition, focusing on Knowledge Media Institute (KMi) news articles. The framework aims for learning and applying lexico-semantic patterns. The extracted information is utilized to populate a knowledge base. Lastly, Capet et al. [3] present a methodology aimed at event extraction for an automated early warning system. The authors employ lexico-semantic patterns for concept matching using dependency chains enhanced using lexicons (word lists), so that concepts are matched whenever syntactically related chains of expressions conveying their constituent concepts occur within the same sentence.

Several advantages stem from the utilization of pattern-based approaches to event extraction. Firstly, pattern-based approaches need less training data than data-driven approaches. Also, it is possible to define powerful expressions by using lexical, syntactical, and semantic elements, and results are easily interpretable and traceable. Patterns are useful when one needs to extract very specific information. However, in order to be able to define patterns that retrieve the correct, desired information, lexical knowledge and possibly also prior domain knowledge is required. Other disadvantages are related to defining and maintaining patterns, as knowledge acquisition is made more difficult (e.g., in costs and consistency) when patterns need to be scaled-up to cover more situations [33] due to the fact that patterns are usually hand-tuned.

2.3 Hybrid Event Extraction

Despite the advantages of both data-driven and knowledge-driven approaches to event extraction, in practice, it is difficult to stay within the boundaries of a single event extraction approach. As both approaches have their disadvantages, combining the two methods could yield the best results. In general, an approach can be viewed as mainly data or knowledge-driven. However, there is an increasing number of researchers that equally combine both approaches, and thus in fact employ hybrid approaches. For instance, it is hard to apply solely pattern-based algorithms successfully, as these algorithms often need for instance bootstrapping or initial clustering, which can be done by means of statistics [29]. Hybrid approaches could emerge when solving the lack of expert knowledge for pattern-based approaches, by applying statistical methods [5]. Also, researchers can combine statistical approaches with (lexical) knowledge, e.g. to prevent unwanted results [28] or to reinforce statistical methods [30].I addition, you can also constrain the learning methods (i.e. data-driven approaches) by using expert knowledge so that a better model is learnt more easily.

In IE literature, many hybrid approaches to text mining are described for extracting events. Most systems are knowledge-driven methods that are aided by data-driven methods, and thus frequently solve the lack of expert knowledge or apply bootstrapping to boost extraction performances, e.g., in terms of precision and recall. For instance, Jungermann and Morik [16] combine lexico-syntactic patterns with conditional random fields (depicted as undirected graphs), in order to extract events from the minutes of plenary sessions of the German parliament. An example of bootstrapping lexical techniques with statistics is given

in [29]. Here, the authors bootstrap a weakly supervised pattern learning algorithm with clusters, in order to be able to extract violelence incidents from online news with high precision and recall, as well as storing these in knowledge bases. Chun et al. [4] extract events from biomedical literature by means of lexico-syntactic patterns, combined with term co-occurrences. Finally, aiming for ontology-based fuzzy event extraction for Chinese e-news summarization, the authors of [18] employ a grammar-based statistical method to text mining, i.e., part-of-speech tagging. However, tagging is based on domain knowledge that is stored in ontologies, thus making the event extraction a hybrid process.

In hybrid event extraction systems, due to the usage of data-driven methods, the amount of required data increases, yet typically remains less than is the case with purely data-driven methods. Compared to a knowledge-driven approach, complexity – and hence required expertise – increases due to the combination of multiple techniques. On the other hand, the amount of expert knowledge that is needed for effective and efficient event discovery is generally less than for pattern-based methods, because of the fact that lack of domain knowledge can be compensated by the use of statistical methods. As for the interpretability, attributing results to specific parts of the event extraction is more difficult due to the addition of data-driven methods. Yet, interpretability still benefits from the use of semantics. Disadvantages of hybrid approaches are mostly related to the multidisciplinary aspects of hybrid systems.

3 Discussion

Table 1 provides a summary of the methods discussed, by combining the results from the discussions in Section 2. Per approach elaborated on in this paper, the employed methods and the type of events that are discovered are summarized. Also, the minimum amount of required data and required domain knowledge and expertise are included, as well as the interpretability of the results.

From the results presented in this table, we derive that in terms of data usage, knowledge-driven event extraction methods require the least amount of data (i.e., experiments are performed on a couple of hundreds of documents or sentences). Data-driven methods on the other hand often make use of more than ten thousand documents. Hybrid methods generally report results on a maximum of ten thousand documents. As for interpretability, i.e., the ease with which the (intermediate) results can be translated to a human-understandable format, data-driven methods perform worst. Knowledge-driven methods on the other hand score higher on interpretability. Especially lexico-semantic pattern approaches have a high level of interpretability, as patterns can easily be translated into natural language, while lexico-syntactic patterns require more effort. Finally, when considering the amount of expert domain knowledge and expertise needed for each approach, data-driven methods require less of both than hybrid and knowledge-driven methods.

As a general guideline for selecting a suitable technique for event extraction, based on the results of our survey, we suggest the usage of knowledge-based

Table 1. Overview of the approaches discussed, displaying the method (*Method*) and the type of events that are discovered (*Events*). Also, the amount of required data (*Data*) is depicted, as well as required domain knowledge and expertise (*Know.* and *Exp.*, respectively), and the interpretability of the results (*Int.*). Note that the reported values in the last four columns are lower bounds.

Technique	Approach	Method	Events	Data	Know.	Exp.	Int.
Data	Okamoto et al. [27]	Hierarchical clustering	Local	Med	Low	Low	Low
	Liu et al. [21]	Graphs, clustering	News	High	Low	Low	Low
	Tanev et al. [34]	Clustering	Violent and disaster news	Med	Low	Low	Low
	Lei et al. [19]	Support Vector Machines	News	High	Low	Low	Low
Knowledge	Nishihara et al. [26]	Lexico-Syntactic	Personal experiences	Low	Med	High	Med
	Aone et al. [1]	Lexico-Syntactic	General	Low	High	High	Med
	Yakushiji et al. [38]	Lexico-Syntactic	Biomedical	Low	Med	High	Med
	Hung et al. [13]	Lexico-Syntactic	Commonsense knowledge	Low	Med	High	Med
	Xu et al. [37]	Lexico-Syntactic	Prize award	Low	Med	High	High
	Li et al. [20]	Lexico-Semantic	Financial	Low	High	High	Med
	Cohen et al. [6]	Lexico-Semantic	Biomedical	Med	High	High	High
	Vargas-Vera et al. [35]	Lexico-Semantic	KMi news	Low	High	High	High
	Capet et al. [3]	Lexico-Semantic	Early warning	Low	High	High	High
Hybrid	Jungermann et al. [16]	Lexico-Syntactic, graphs	German parliament	Med	Med	High	Med
	Piskorski et al. [29]	Lexico-Semantic, clustering	Violent news	High	Med	Med	Med
	Chun et al. [4]	Lexico-Syntactic, co-occurences	Biomedical	Med	Med	Med	Med
	Lee et al. [18]	Ontology-based Part-Of-Speech tagging	Chinese news	N/A	Med	Med	Low

techniques for casual users (e.g., students) that prefer an interactive, query-driven approach to event extraction, assuming domain knowledge and expertise to be readily available. Users can easily specify patterns in a language that is close to their own natural language, without being bothered with statistical details and model fine-tuning. On the other hand, users like (academic) researchers would benefit from both hybrid and data-driven approaches, as these are less restricted by, for example, grammars.

4 Conclusions

In this paper, we investigated the main approaches to event extraction from text that are elaborated on in the current body of literature. Overall, data-driven methods require many data and little domain knowledge and expertise, while having a low interpretability. Conversely, for knowledge-based event extraction little data is required, but domain knowledge and expertise is needed. These approaches generally offer a higher traceability of the results. Finally, hybrid approaches seem to be a compromise between data and knowledge-driven approaches, requiring a medium amount of data and domain knowledge and offering medium interpretability. However, it should be noted that the amount of expertise needed is high, due to the fact that multiple techniques are combined. As a guideline, we advise knowledge-driven techniques for casual and novice users, whereas data-driven are more suitable for advanced users.

References

1. Aone, C., Ramos-Santacruz, M.: REES: A Large-Scale Relation and Event Extraction System. In: 6th Applied Natural Language Processing Conference (ANLP 2000). pp. 76–83. Association for Computational Linguistics (2000)
2. Borsje, J., Hogenboom, F., Frasincar, F.: Semi-Automatic Financial Events Discovery Based on Lexico-Semantic Patterns. International Journal of Web Engineering and Technology 6(2), 115–140 (2010)
3. Capet, P., Delavallade, T., Nakamura, T., Sandor, A., Tarsitano, C., Voyatzi, S.: Intelligent Information Processing IV, IFIP International Federation for Information Processing, vol. 288, chap. A Risk Assessment System with Automatic Extraction of Event Types, pp. 220–229. Springer Boston (2008)
4. Chun, H.W., Hwang, Y.S., Rim, H.C.: Unsupervised Event Extraction from Biomedical Literature Using Co-occurrence Information and Basic Patterns. In: 1st International Joint Conference on Natural Language Processing (IJCNLP 2004). Lecture Notes in Computer Science, vol. 3248, pp. 777–786. Springer-Verlag Berlin Heidelberg (2004)
5. Cimiano, P., Staab, S.: Learning by Googling. SIGKDD Explorations Newsletter 6(2), 24–33 (2004)
6. Cohen, K.B., Verspoor, K., Johnson, H.L., Roeder, C., Ogren, P.V., Baumgartner, Jr., W.A., White, E., Tipney, H., Hunter, L.: High-Precision Biological Event Extraction with a Concept Recognizer. In: Workshop on BioNLP: Shared Task collocated with the NAACL-HLT 2009 Meeting. pp. 50–58. Association for Computational Linguistics (2009)

7. Cunningham, H.: GATE, a General Architecture for Text Engineering. Computers and the Humanities 36(2), 223–254 (2002)
8. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002). pp. 168–175. Association for Computational Linguistics (2002)
9. Ferrucci, D., Lally, A.: UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. Natural Language Engineering 10(3–4), 327–348 (2004)
10. Frasincar, F., Borsje, J., Levering, L.: A Semantic Web-Based Approach for Building Personalized News Services. International Journal of E-Business Research 5(3), 35–53 (2009)
11. Hearst, M.A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: 14th Conference on Computational Linguistics (COLING 1992). vol. 2, pp. 539–545 (1992)
12. Hearst, M.A.: WordNet: An Electronic Lexical Database and Some of its Applications, chap. Automated Discovery of WordNet Relations, pp. 131–151. MIT Press (1998)
13. Hung, S.H., Lin, C.H., Hong, J.S.: Web Mining for Event-Based Commonsense Knowledge Using Lexico-Syntactic Pattern Matching and Semantic Role Labeling. Expert Systems with Applications 37(1), 341–347 (2010)
14. Ikenberry, D.L., Ramnath, S.: Underreaction to Self-selected News Events: The Case of Stock Splits. Review of Financial Studies 15(2), 489–526 (2002)
15. Ikonomakis, M., Kotsiantis, S., Tampakas, V.: Text Classification Using Machine Learning Techniques. WSEAS Transactions on Computers 4(8), 966–974 (2005)
16. Jungermann, F., Morik, K.: Enhanced Services for Targeted Information Retrieval by Event Extraction and Data Mining. In: 13th International Conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems (NLDB 2008). Lecture Notes in Computer Science, vol. 5039, pp. 335–336. Springer-Verlag Berlin Heidelberg (2008)
17. Kamijo, S., Matsushita, Y., Ikeuchi, K., Sakauchi, M.: Traffic monitoring and accident detection at intersections. IEEE Transactions on Intelligent Transportation Systems 1(2), 108–118 (2000)
18. Lee, C.S., Chen, Y.J., Jian, Z.W.: Ontology-Based Fuzzy Event Extraction Agent for Chinese E-News Summarization. Expert Systems with Applications 25(3), 431–447 (2003)
19. Lei, Z., Wu, L.D., Zhang, Y., Liu, Y.C.: A System for Detecting and Tracking Internet News Event. In: 6th Pacific-Rim Conference on Multimedia (PCM 2005). Lecture Notes in Computer Science, vol. 3767, pp. 754–764. Springer-Verlag Berlin Heidelberg (2005)
20. Li, F., Sheng, H., Zhang, D.: Event Pattern Discovery from the Stock Market Bulletin. In: 5th International Conference on Discovery Science (DS 2002). Lecture Notes in Computer Science, vol. 2534, pp. 35–49. Springer-Verlag Berlin Heidelberg (2002)
21. Liu, M., Liu, Y., Xiang, L., Chen, X., Yang, Q.: Extracting Key Entities and Significant Events from Online Daily News. In: 9th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2008). Lecture Notes in Computer Science, vol. 5326, pp. 201–209. Springer-Verlag Berlin Heidelberg (2008)
22. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, 1st edn. (1999)

23. Michaely, R., Thaler, R.H., Womack, K.L.: Price Reactions to Dividend Initiations and Omissions: Overreaction or Drift? Journal of Finance 50(2), 573–608 (1995)
24. Mitchell, M.L., Mulherin, J.H.: The Impact of Public Information on the Stock Market. Journal of Finance 49(3), 923–950 (1994)
25. Moreau, N.: Best Practices in Language Resources for Multilingual Information Access. Tech. rep., TrebleCLEF Consortium (2009), From: `http://www.trebleclef.eu/getfile.php?id=255`
26. Nishihara, Y., Sato, K., Sunayama, W.: Event Extraction and Visualization for Obtaining Personal Experiences from Blogs. In: Symposium on Human Interface 2009 on Human Interface and the Management of Information. Information and Interaction. Part II. Lecture Notes in Computer Science, vol. 5618, pp. 315–324. Springer-Verlag Berlin Heidelberg (2009)
27. Okamoto, M., Kikuchi, M.: Discovering Volatile Events in Your Neighborhood: Local-Area Topic Extraction from Blog Entries. In: 5th Asia Information Retrieval Symposium (AIRS 2009). Lecture Notes in Computer Science, vol. 5839, pp. 181–192. Springer-Verlag Berlin Heidelberg (2009)
28. Pakhomov, S.: Semi-Supervised Maximum Entropy Based Approach to Acronym and Abbreviation Normalization in Medical Texts. In: 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002). pp. 160–167. Association for Computational Linguistics (2002)
29. Piskorski, J., Tanev, H., Wennerberg, P.O.: Extracting Violent Events From On-Line News for Ontology Population. In: 10th International Conference on Business Information Systems (BIS 2007). Lecture Notes in Computer Science, vol. 4439, pp. 287–300. Springer-Verlag Berlin Heidelberg (2007)
30. Punyakanok, V., Roth, D., Yih, W.: The Importance of Syntactic Parsing and Inference in Semantic Role Labeling. Computational Linguistics 34(2), 257–287 (2008)
31. Rosen, R.J.: Merger Momentum and Investor Sentiment: The Stock Market Reaction to Merger Announcements. Journal of Business 79(2), 987–1017 (2006)
32. Schouten, K., Ruijgrok, P., Borsje, J., Frasincar, F., Levering, L., Hogenboom, F.: A Semantic Web-Based Approach for Personalizing News. In: 25th Symposium On Applied Computing (SAC 2010). pp. 854–861. ACM (2010)
33. Su, K.Y., Chiang, T.H., Chang, J.S.: An Overview of Corpus-Based Statistics-Oriented (CBSO) Techniques for Natural Language Processing. Computational Linguistics and Chinese Language Processing 1(1), 101–157 (1996)
34. Tanev, H., Piskorski, J., Atkinson, M.: Real-Time News Event Extraction for Global Crisis Monitoring. In: 13th International Conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems (NLDB 2008). Lecture Notes in Computer Science, vol. 5039, pp. 207–218. Springer-Verlag Berlin Heidelberg (2008)
35. Vargas-Vera, M., Celjuska, D.: Event Recognition on News Stories and Semi-Automatic Population of an Ontology. In: 3rd IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004). pp. 615–618 (2004)
36. Wei, C.P., Lee, Y.H.: Event detection from Online News Documents for Supporting Environmental Scanning. Decision Support Systems 36(4), 385–401 (2004)
37. Xu, F., Uszkoreit, H., Li, H.: Automatic Event and Relation Detection with Seeds of Varying Complexity. In: AAAI Workshop on Event Extraction and Synthesis (2006)
38. Yakushiji, A., Tateisi, Y., Miyao, Y.: Event Extraction from Biomedical Papers using a Full Parser. In: 6th Pacific Symposium on Biocomputing (PSB 2001). pp. 408–419 (2001)