

Leveraging Multilingual Training for Limited Resource Event Extraction

Andrew Hsi Yiming Yang Jaime Carbonell Ruochen Xu

Carnegie Mellon University
Pittsburgh, PA 15213 USA

{ahsi, yiming, jgc, ruochenx}@cs.cmu.edu

Abstract

Event extraction has become one of the most important topics in information extraction, but to date, there is very limited work on leveraging cross-lingual training to boost performance. We propose a new event extraction approach that trains on multiple languages using a combination of both language-dependent and language-independent features, with particular focus on the case where target domain training data is of very limited size. We show empirically that multilingual training can boost performance for the tasks of event trigger extraction and event argument extraction on the Chinese ACE 2005 dataset.

1 Introduction

Traditionally, event extraction has focused on monolingual training – typically English (Grishman et al., 2005; Ji and Grishman, 2008; Gupta and Ji, 2009; Liao and Grishman, 2010; Liao and Grishman, 2011; Li et al., 2013; Bronstein et al., 2015), and occasionally Chinese or other languages (Chen and Ji, 2009b; Piskorski et al., 2011; Li et al., 2012; Chen and Ng, 2012; Chen and Ng, 2014). However, apart from a few isolated studies (Chen and Ji, 2009a; Piskorski et al., 2011), to date there is very little work leveraging cross-lingual information for event extraction. Cross-lingual approaches have proven useful for many other tasks in natural language processing (NLP), including part-of-speech (POS) tagging (Snyder et al., 2009; Cohen et al., 2011), dependency parsing (Zeman and Resnik, 2008; Cohen et al., 2011; McDonald et al., 2011; Ammar et al., 2016), and named entity recognition (Richman and Schone, 2008).

An important issue in event extraction is that the amount of available training data is often insufficient or unbalanced across domains and/or languages. Event extraction training datasets typically contain merely a few hundreds of documents, owing to the complexity and high costs of human annotation. This issue is even more severe for new event types in new languages. This provides strong motivation to leverage existing language resources for event extraction in new languages. This problem is closely related to low-resource NLP, which has been gathering increased interest among researchers (Garrette et al., 2013; Miao et al., 2013; Duong et al., 2014; Duong et al., 2015).

In this paper, we propose a novel approach for cross-lingual event extraction, which trains on multiple languages and leverages both language-dependent and language-independent features in order to boost performance. Using such a system we aim to jointly leverage available multilingual resources (annotated data and induced features) to overcome the annotation-scarcity issue in the target language of interest. Empirically we show that our approach can substantially improve the performance of monolingual systems for the task of Chinese event argument extraction. Our approach is novel compared to existing work in that we have no reliance on using either high quality machine translation or manually aligned documents, which may be unavailable for a given target language.

The rest of the paper is organized as follows. Section 2 introduces relevant terminology used in the event extraction field. Section 3 describes some related work on event extraction and cross-lingual NLP. Section 4 details our event extraction system and the types of features we use. In Section 5, we describe

our experimental setup and discuss results for both cross-lingual event trigger extraction and cross-lingual event argument extraction. We conclude in Section 6 with some ideas for future work.

2 Terminology and Task Definitions

We will begin by briefly introducing the basic terminology used in the event extraction field and the task definitions by the Automatic Content Extraction (ACE) Evaluation program¹ conducted by the National Institute of Science and Technology (NIST). The ACE program focused on entity detection, relation detection, and event detection – among these, we focus in this paper specifically on the event detection task, which consists of event trigger extraction and event argument extraction.

- An *event* is something that happens at a particular time and place, often involving one or more people. Examples include births, attacks, and arrests.
- An *event mention* is a particular textual occurrence of an event. A text may contain several different mentions that all refer to the same physical event.
- An *event trigger* is the specific word in a sentence that indicates the existence of an event.
- An *event argument* is an entity fulfilling a specific role within the event. The set of permissible roles depends on the event type. For example, the Attacker role would be valid for a Conflict.Attack event, while the same role would be invalid for an event of type Business.Declare-Bankruptcy. Additionally, all event types in ACE include roles for Time and Place.
- Lastly, an *event argument mention* is a particular textual occurrence of an event argument.

The *event trigger extraction task* is to identify all of the event triggers contained within a set of documents. The *event argument extraction task* is to identify all of the event arguments contained within a set of documents. In most cases, the event trigger extraction step is conducted first to identify the event mentions, and then event argument extraction is performed on top of this to identify the particular entities fulfilling argument roles for these event mentions.

3 Related Work

A variety of machine learning methods have been used for event extraction in the past, including pipelines of classifiers (Grishman et al., 2005; Ji and Grishman, 2008; Liao and Grishman, 2011), joint inference models (Li et al., 2013; Li and Ji, 2014; Yang and Mitchell, 2016), and neural networks (Nguyen and Grishman, 2015; Chen et al., 2015) – the vast majority of which focus solely on the English monolingual training scenario. A subset of the event extraction literature has considered the study of Chinese event extraction (Chen and Ji, 2009b; Li et al., 2012; Chen and Ng, 2012; Chen and Ng, 2014). However, most of these works also focus solely on the monolingual case, and do not leverage any additional training data from other languages.

The most related work to our approach is that of Chen and Ji (2009a). In their model, they designed a co-training approach to augment a small Chinese training corpus with additional examples from an unlabeled corpus. Given a parallel corpus of English-Chinese documents and a monolingual English event extraction system (trained on annotated English documents), they used the system to predict the event labels on the English part of the parallel documents and project the predicted labels to the Chinese part of the parallel corpus based on gold standard alignments. The Chinese system is then trained using a combination of the originally annotated Chinese document and the parallel texts with the projected labels. This approach offered slight improvements in the event trigger extraction task and the event argument extraction task (see Section 2 for definitions), but relies on having in-domain parallel texts either aligned by humans or by high quality machine translation models between the source and target languages. In contrast, our proposed approach has no such limitation, and hence is easier to apply to any target language of interest.

¹<http://www.itl.nist.gov/iad/mig/tests/ace/>

Another related work is that of Piskorski et al. (2011), who use cross-lingual information to refine the results of event extraction. In particular, they run several monolingual event extraction systems independently, translate the extracted argument fillers into English, and merge together argument fillers across documents. Using this cross-document information fusion, they find improved performance over monolingual systems. However, this work relies on having documents across multiple languages that describe the exact same event, which is an unrealistic case in practice. Additionally, they also rely on having high quality machine translation in order to translate the argument output of each monolingual system into English.

There does exist some prior work on the broader field of cross-lingual information extraction. Riloff et al. (2002) start with English annotated source texts, create a parallel corpus via machine translation, and project the annotations via alignments. The projected annotations are then used to conduct training in the target language. Sudo et al. (2004) presents an approach for extracted patterns in a source language and translating these patterns for use on a target language. However, these works are limited to entity extraction, whereas our focus is on event extraction. Furthermore, both works rely on having high-quality machine translation output.

Beyond information extraction, cross-lingual training has offered benefits for a variety of tasks. McDonald et al. (2011) use a delexicalized English parser to seed a lexicalized parser in the target language, and then iteratively improve upon this model via constraint driven learning. Duong et al. (2014) develop a POS tagger for low resource languages by first projecting predicted English POS tags across parallel data to obtain target language training data, and then further augment this with a small amount of annotated data in the target language. Ammar et al. (2016) developed a language universal dependency parser by using language-independent features to create a general model, and fine-tuning the resulting model with language-specific features and embeddings. Similarly to our model, this method has no requirement about the availability of alignments and parallel text.

4 System Description

To date, there exist only a handful of languages that have ACE-style event annotations, yet this leaves a vast number of languages in which people have no capacity to conduct event extraction. This problem is compounded by the difficulty of event extraction annotation. Annotation of documents for event extraction is a very labor-intensive, costly task – even the standard benchmark dataset of ACE 2005 only contains several hundred annotated English documents. It is inconceivable to believe that we will ever have similar datasets for every language of interest.

A natural effort therefore, is to leverage as much information as we can from existing “high-resource” event-extraction languages, along with whatever limited training data we may have for the target language. To accomplish this, we create a standard pipeline-of-classifiers approach to event extraction, and then augment this model with multilingual features.

The overall system architecture may be seen in Figure 1. We begin by preprocessing the data to obtain tokenizations, POS tags, and dependency parses. We then extract our trigger features, and run a multi-class classifier to predict the trigger labels. We subsequently extract our argument features using the original preprocessed data in combination with the system predicted trigger labels, and run a second multi-class classifier to make predictions on the argument roles.

4.1 Trigger Prediction

We begin by describing our trigger prediction component. For the task of event trigger prediction, we train a multi-class logistic regression classifier using LIBLINEAR (Fan et al., 2008). For each word, we make a prediction on the event trigger type – one of 33 given types from the ACE ontology, or the NONE category to represent when a word does not trigger an ACE event.

The trigger system uses a variety of monolingual features, seen in Table 1. For the majority of the features, we use binary indicator functions to represent whether the feature is either active or inactive for the particular data instance. For the word embedding features, we use the real-valued vectors directly for representing each word.

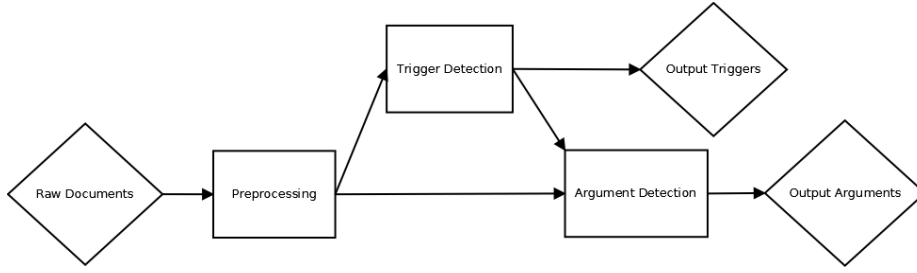


Figure 1: Architecture for our event extraction system. The argument component relies on the predictions from the trigger component.

Event Trigger Extraction Features
Lexical features (e.g. words and lemmas within a context window)
Length of the current word
Language-specific POS tags within a context window
Universal POS tags within a context window
Word embedding vector for current word
Dependent/Governor information from dependency parsing
Bilingual dictionary word pairs

Table 1: Features used in the Event Trigger Extraction component

Multilingual training is leveraged via the use of four types of features: 1.) Universal POS Tags (Petrov et al., 2012), 2.) Universal Dependencies (McDonald et al., 2013), 3.) limited bilingual dictionaries, and 4.) aligned multilingual word embeddings. The Universal POS tags and Universal Dependencies allow us to use a single set of tags for both languages, which thereby enables the use of English training data directly in our model. The bilingual dictionary provides a limited number of translations between words, and may be used both directly in the model and for aligning word embeddings. The aligned word embeddings similarly allow us to directly use English training data, as each component in the vector is aligned to semantically match those of the target word embeddings.

To obtain aligned word embeddings, we first start with monolingual word embeddings. We obtain monolingual texts for both English and the target language from Wikipedia, and independently train word embeddings for each language using word2vec (Mikolov et al., 2013). These monolingual embeddings are then aligned by solving a regression problem.

Let $D = (x_i, z_i)_{i=1}^n$ represent a limited bilingual dictionary between the two languages, where $x_i \in \mathbb{R}^{d_1}$ is the word embedding of word i in English and $z_i \in \mathbb{R}^{d_2}$ is the word embedding of its translation in the target language. Our regression problem is to find a transformation matrix W minimizing the following objective function:

$$\min_{W \in \mathbb{R}^{d_2 \times d_1}} \sum_{i=1}^n \|Wx_i - z_i\|^2 + \lambda \|W\|_F^2 \quad (1)$$

The first term of the objective function serves to ensure that the projected vectors in English closely match those of their translations in the target language. The second term is a regularization term to avoid overfitting. This problem has a closed form solution, which is given by:

$$W^* = ZX^T (XX^T + \lambda I)^{-1} \quad (2)$$

where $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d_1 \times n}$ and $Z = [z_1, z_2, \dots, z_n] \in \mathbb{R}^{d_2 \times n}$

The resulting aligned embeddings may then be used in our feature set. Using these aligned embeddings is preferable over just using the direct dictionary translations, as many words in both the source and target language may not appear in the bilingual dictionary. In our approach, once a mapping is found between two embedded spaces, we may project any word into this shared space.

4.2 Argument Prediction

We now describe the argument prediction component of our system. For this component, we require a few additional fields of information: 1.) event trigger words, and 2.) entity mentions. Event trigger words and entity mentions may be provided either as gold information or extracted automatically using machine learning approaches.

In each sentence, for each (trigger word, entity mention) pair, we make a prediction on the argument role (if any) between the trigger and the entity. The ACE ontology contains 35 different argument types, and we also include the NONE label to indicate when there is no relationship between the trigger and the entity. As in the previous case of trigger extraction, we once again accomplish this by training a multi-class logistic regression classifier using LIBLINEAR.

Event Argument Extraction Features
Lexical features about the entity phrase
Lexical features for individual words in the entity phrase
Entity type, subtype
Event type and subtype of trigger word
Existence of any other candidate entities in the same sentence
Distance between the trigger and entity
Dependent/Governor information from dependency parsing
Bilingual dictionary word pairs

Table 2: Features used in the Event Trigger Extraction component

Features for the event argument extraction component may be seen in Table 2. Multilingual information is leveraged in a similar way to the trigger prediction component, using Universal POS tags, Universal Dependencies, and any available bilingual dictionaries to learn from English training data.

5 Experimental Setup

To test our approach, we conduct experiments on two separate tasks: event trigger extraction and event argument extraction. We begin by describing our experimental setup and metrics, and subsequently show empirical results on the two tasks.

5.1 Dataset

We conduct experiments on the ACE 2005 dataset, the most dominating benchmark dataset for event trigger extraction and event argument extraction. The English and Chinese portions of ACE each contain several hundred documents annotated with gold standard entity and event information. We preprocess the raw text of each document using Stanford CoreNLP (Manning et al., 2014). We split the Chinese portion into 10 folds, and perform cross-validation. In the ACE collection, the number of labeled Chinese documents is approximately the same as the number of English documents, so to simulate a low resource scenario, we select just one training fold for each round of cross-validation. We use another fold for parameter tuning, and use the remaining folds in each round for testing. We use CC-CEDICT² as our bilingual dictionary between English and Chinese.

For our baseline system, we use just the Chinese data for training, and only the monolingual features. Our cross-lingual system uses the entire set of features, and additionally incorporates the entire English portion of ACE 2005 for training.

5.2 Metrics

We report both micro-averaged and macro-averaged precision, recall, and F1. Typically the event extraction community reports micro-averaged results, which give the overall performance after pooling all the labels together. However, we argue that only presenting this single perspective provides a skewed

²downloadable from <https://cc-cedict.org/wiki/start>

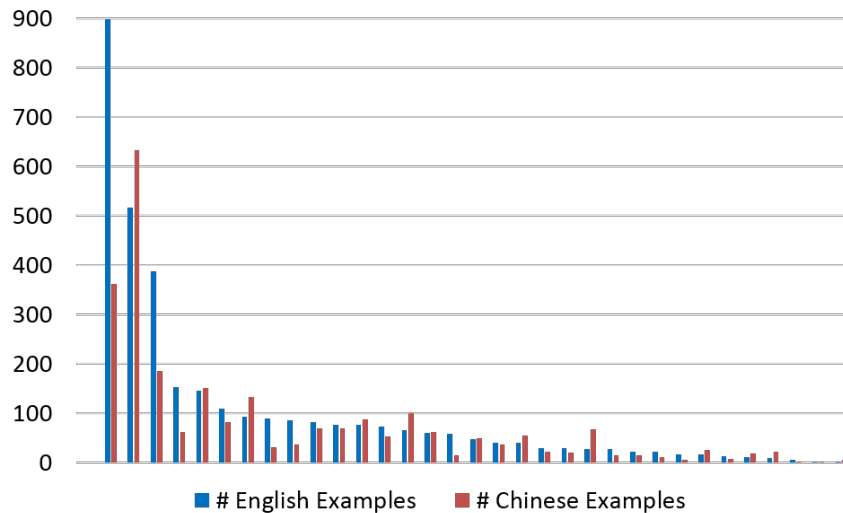


Figure 2: Distribution of Trigger Types in ACE 2005. Each bar represents one of the event types found in ACE 2005, and the height of the bar represents the number of event mentions for said class.

Trigger Types (frequency)	
Conflict.Attack (1252)	Movement.Transport (607)
Life.Die (488)	Personnel.End-Position (196)
Contact.Meet (190)	Personnel.Elect (143)
Transaction.Transfer-Money (140)	Life.Injure (116)
Justice.Charge-Indict (107)	Contact.Phone-Write (107)
Transaction.Transfer-Ownership (101)	Justice.Trial-Hearing (100)
Justice.Sentence (94)	Personnel.Start-Position (92)
Justice.Arrest-Jail (88)	Justice.Convict (75)
Conflict.Demonstrate (72)	Life.Marry (58)
Justice.Sue (55)	Life.Be-Born (46)
Business.Declare-Bankruptcy (40)	Justice.Appeal (39)
Business.Start-Org (38)	Justice.Release-Parole (35)
Business.End-Org (33)	Life.Divorce (28)
Justice.Fine (28)	Justice.Execute (20)
Business.Merge-Org (18)	Personnel.Nominate (11)
Justice.Acquit (7)	Justice.Extradite (3)
Justice.Pardon (2)	

Table 3: Counts of trigger types in English ACE 2005.

view of the system performance, as this type of measure is highly favorable to the majority class labels. This is particularly an issue for the ACE 2005 collection, as the distribution over both trigger types and argument roles is highly skewed, as seen in Figures 2 and 3.

Table 3 shows the specific counts for each trigger type in the English portion of ACE 2005. The trigger type “Conflict.Attack” occurs far more frequently in the texts than any of the others – more than twice that of the second most common type. On the other extreme, the highly infrequent types only occur very rarely in the text. Analysis of the argument counts (Table 4) shows a similar situation. While not as badly skewed as in the trigger case, there is still noticeable disparity between the most frequent and least frequent classes. Some of this may be attributed to the fact that ACE includes several argument types that correspond to different varieties of “Time”, but even if we ignore the “Time”-type arguments, there are still nine argument classes with less than 100 examples each.

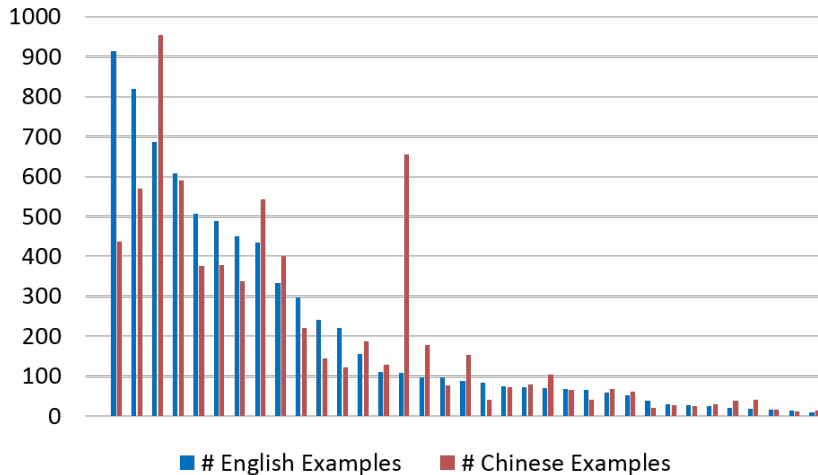


Figure 3: Distribution of Argument Roles in ACE 2005. Each bar represents one of the argument roles found in ACE 2005, and the height of the bar represents the number of argument mentions for said class.

Argument Roles (frequency)		
Person (1064)	Place (891)	Time-Within (699)
Entity (685)	Attacker (564)	Target (535)
Victim (529)	Destination (462)	Agent (368)
Defendant (359)	Crime (245)	Instrument (244)
Origin (160)	Artifact (131)	Position (111)
Giver (108)	Recipient (107)	Adjudicator (106)
Org (105)	Buyer (79)	Vehicle (78)
Money (75)	Sentence (74)	Plaintiff (72)
Time-Holds (68)	Time-Starting (52)	Beneficiary (42)
Seller (37)	Prosecutor (29)	Time-After (24)
Time-Before (20)	Time-Ending (19)	Time-At-End (15)
Time-At-Beginning (15)	Price (8)	

Table 4: Counts of the argument roles in English ACE 2005.

5.3 Event Trigger Extraction results

Results for trigger extraction may be seen in Table 5. The cross-lingual approach shows improved performance on both the micro-averaged and macro-averaged F1 metrics, demonstrating the success of incorporating multilingual training. On the macro-averaged metric, we see an improvement of 10.7%, and on the micro-averaged metric, an improvement of 3.9%. We find these improvements on F1 to be significant under a t-test with $\alpha=0.01$.

As one would expect, the macro-averaged scores are noticeably lower than the micro-averaged scores, which suggests that the rare classes for event triggers suffer from worse performance than the frequent classes. Note that the difference in performance between the two approaches is larger on the macro-average metric. This suggests that the addition of multilingual training is playing a key role to improve performance on these particular rare classes.

	Macro-Average			Micro-Average		
	Precision	Recall	F1	Precision	Recall	F1
Monolingual approach	0.421	0.183	0.233	0.646	0.271	0.381
Cross-lingual approach	0.443	0.209	0.258	0.635	0.288	0.396

Table 5: Results of Trigger Extraction Task on Chinese ACE 2005

5.4 Event Argument Extraction results

Results for argument extraction may be seen in Tables 6 and 7. Table 6 shows the optimal performance obtained by using the gold event triggers as input, and Table 7 shows the more realistic scenario of using the system predicted triggers as input. In both cases, we utilize the existing gold entity information provided by the ACE collection.

We see even larger boosts in macro-average performance from the argument extraction component than from the trigger extraction component – using gold triggers we get a 34.8% improvement on macro F1, and using system predicted triggers we get a 28.2% improvement on macro F1. On micro-average metrics, we find a smaller, but still meaningful boost in performance: 3.2% improvement on micro F1 when using gold triggers as input, and 5.7% improvement on micro F1 when using the system predicted triggers. We find all of our argument results to show significant improvements on F1 over the monolingual equivalents under a t-test with $\alpha=0.01$.

We suspect that the noticeably larger gains in argument macro-average performance compared to trigger performance may be due to the more semantic nature of the task. Trigger words are primarily dependent on lexical information, whereas arguments rely more heavily on deeper semantic information such as that provided by dependency parsing. Information leveraged from sources like Universal Dependencies is therefore likely to have a greater effect in the argument extraction setting, and in particular on the rare classes that do not have enough data to perform well under monolingual training.

	Macro-Average			Micro-Average		
	Precision	Recall	F1	Precision	Recall	F1
Monolingual approach	0.510	0.189	0.250	0.744	0.336	0.462
Cross-lingual approach	0.556	0.267	0.337	0.731	0.355	0.477

Table 6: Results of Argument Extraction Task on Chinese ACE 2005, using gold trigger labels as input

	Macro-Average			Micro-Average		
	Precision	Recall	F1	Precision	Recall	F1
Monolingual approach	0.400	0.080	0.124	0.651	0.140	0.230
Cross-lingual approach	0.422	0.105	0.159	0.651	0.150	0.243

Table 7: Results of Argument Extraction Task on Chinese ACE 2005, using system predicted trigger labels as input

6 Conclusion

In this paper, we proposed a cross-lingual approach to event extraction that leverages both language-dependent and language-independent features to train with multiple languages. Motivated by the necessity of developing new techniques for expansion of event extraction to new languages, and inspired by the recent success stories of cross-lingual NLP, we developed an approach which allows us to incorporate any additional training data from other languages, while also maximally utilizing whatever monolingual data we have available. Our experimental results show improved performance for event trigger extraction and event argument extraction with multilingual training, under both the macro-averaged and micro-averaged metrics. These are very encouraging numbers, and we believe this indicates event extraction to be a promising direction for future cross-lingual researchers to explore.

There are several natural extensions to this work. One interesting direction of research would be to explore an actual (rather than simulated) low-resource language, where the target language may not only have little (or no) training data, but may not even have available tools for preprocessing tasks (POS tagging, entity recognition, parsing, etc.). This is an important area of research, as the majority of the world’s languages do not have such tools available. A second direction of promising research is consider the case of not just leveraging a single source language, but to rather include multiple source languages.

A final interesting direction is to adapt recent neural methods for cross-lingual NLP, such as those by Ammar et al. (2016). By using a more sophisticated machine learning approach, we may be able to improve our multilingual efforts even further than our current approach.

Acknowledgements

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

References

- Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. In *TACL*.
- Ofer Bronstein, Ido Dagan, Qi Li, Heng Ji, and Anette Frank. 2015. Seed-based event trigger labeling: How far can event descriptions get us? In *ACL*.
- Zheng Chen and Heng Ji. 2009a. Can one language bootstrap the other: A case study on event extraction. In *NAACL HLT Workshop on Semi-supervised Learning for Natural Language Processing*.
- Zheng Chen and Heng Ji. 2009b. Language specific issue and feature exploration in chinese event extraction. In *HLT-NAACL*.
- Chen Chen and Vincent Ng. 2012. Joint modeling for chinese event extraction with rich linguistic features. In *COLING*.
- Chen Chen and Vincent Ng. 2014. Sinocoreferencer: An end-to-end chinese event coreference resolver. In *LREC*.
- Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL*.
- Shay B. Cohen, Dipanjan Das, and Noah A. Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *EMNLP*.
- Long Duong, Trevor Cohn, Karin Verspoor, Steven Bird, and Paul Cook. 2014. What can we get from 1000 tokens? a case study of multilingual pos tagging for resource-poor languages. In *EMNLP*.
- Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. A neural network model for low-resource universal dependency parsing. In *EMNLP*.
- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. In *JMLR*.
- Dan Garrette, Jason Mielens, and Jason Baldridge. 2013. Real-world semi-supervised learning of pos-taggers for low-resource languages. In *ACL*.
- Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nys english ace 2005 system description. In *Proc. ACE 2005 Evaluation Workshop*.
- Prashant Gupta and Heng Ji. 2009. Predicting unknown time arguments based on cross-event propagation. In *ACL-IJCNLP*.
- Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*.
- Qi Li and Heng Ji. 2014. Constructing information networks using one single model. In *EMNLP*.
- Peifeng Li, Guodong Zhou, Qiaoming Zhu, and Libin Hou. 2012. Employing compositional semantics and discourse consistency in chinese event extraction. In *EMNLP*.
- Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *ACL*.
- Shasha Liao and Ralph Grishman. 2010. Filtered ranking for bootstrapping in event extraction. In *ACL*.
- Shasha Liao and Ralph Grishman. 2011. Acquiring topic features to improve event extraction: in pre-selected and balanced collections. In *RANLP*.

- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL*.
- Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *EMNLP*.
- Ryan McDonald, Joakim Nivre, Yvonne Quirnbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *ACL*.
- Yajie Miao, Florian Metze, and Shourabh Rawat. 2013. Deep maxout networks for low-resource speech recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*.
- Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *ACL*.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *LREC*.
- Jakub Piskorski, Jenya Belayeva, and Martin Atkinson. 2011. Exploring the usefulness of cross-lingual information fusion for refining real-time news event extraction: A preliminary study. In *RANLP*.
- Alexander E. Richman and Patrick Schone. 2008. Mining wiki resources for multilingual named entity recognition. In *ACL*.
- Ellen Riloff, Charles Schafer, and David Yarowski. 2002. Inducing information extraction systems for new languages via cross-language projection. In *COLING*.
- Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. 2009. Adding more languages improves unsupervised multilingual part-of-speech tagging: A bayesian non-parametric approach. In *NAACL*.
- Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2004. Cross-lingual information extraction system evaluation. In *COLING*.
- Bishan Yang and Tom Mitchell. 2016. Joint extraction of events and entities within a document context. In *NAACL*.
- Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *IJCNLP*.