Tunable Domain-Independent Event Extraction in the MIRA Framework

Georgi Georgiev[1]
georgi.georgiev@ontotext.com

Kuzman Ganchev[1]
kuzman.ganchev@ontotext.com

Vassil Momtchev[1]
vassil.momtchev@ontotext.com

Deyan Peychev[1]
deyan.peychev@ontotext.com

Preslav Nakov[1]
preslav.nakov@ontotext.com

Angus Roberts[2]
a.roberts@dcs.shef.ac.uk

[1] Ontotext AD, 135 Tsarigradsko Chaussee, Sofia 1784, Bulgaria
[2] The Department of Computer Science, Regent Court 211 Portobello, Sheffield, S1 4DP. UK.

Abstract

We describe the system of the PIKB team for BioNLP'09 Shared Task 1, which targets tunable domain-independent event extraction. Our approach is based on a three-stage classification: (1) trigger word tagging, (2) simple event extraction, and (3) complex event extraction. We use the MIRA framework for all three stages, which allows us to trade precision for increased recall by appropriately changing the loss function during training. We report results for three systems focusing on recall (R = 28.88%), precision (P = 65.58%), and F_1-measure (F_1 = 33.57%), respectively.

1 Introduction

Molecular interactions have been the focus of intensive research in the development of in-silico biology. Recent developments like the *Pathway and Interaction Knowledge Base* (*PIKB*) aim to make available to the user the large semantics of the existing molecular interactions data using massive knowledge syndication. PIKB is part of *LinkedLifeData*[1], a platform for semantic data integration based on RDF[2] syndication and lightweight reasoning.

Our system is based on the MIRA framework where, by appropriately changing the loss function on training, we can achieve any desirable balance between precision and recall. For example, low precision with high recall would be appropriate in a search that aims to identify as many potential candidates as possible to be further examined by the user, while high precision might be essential when adding relations to a knowledge base. Such a tunable system is practical for a variety of important tasks, including but not limited to, populating extracted facts in PIKB and reasoning on top of new and old data.

Our system is based on a three-stage classification process: (1) trigger word tagging using a linear sequence model, (2) simple event extraction, and (3) complex event extraction. In stage (2), we generate relations between a trigger word and one or more proteins, while in stage (3), we look for complex interactions between simple events, trigger words and proteins. We use MIRA for all three stages with a loss function tuned for high recall.

2 One-best MIRA and Loss Functions

In what follows, x_i will denote a generic input sentence, and y_i will be the "gold" labeling of x_i. For each pair of a sentence x_i and a labeling y, we compute a vector-valued feature representation $f(x_i, y)$. Given a weight vector w, the dot-product $w \cdot f(x, y)$ ranks the possible labelings y of x; we will denote the top scoring labeling as $y_w(x)$. As with hidden Markov models (Rabiner, 1989), $y_w(x)$ can be computed efficiently for suitable feature functions using dynamic programming.

The learning portion of our method requires finding a weight vector w that scores the correct labeling of the training data higher than any incorrect labeling. We used a one-best version of MIRA (Crammer, 2004; McDonald et al., 2005) to choose w. MIRA is an online learning algorithm that updates the weight vector w for each training sentence x_i according to the following rule:

[1] *http://www.linkedlifedata.com*
[2] *http://www.w3.org/RDF/*

Proceedings of the Workshop on BioNLP: Shared Task, pages 95–98,
Boulder, Colorado, June 2009.

$$w_{\text{new}} = \arg\min_{w} \|w - w_{\text{old}}\|$$
$$\text{s.t. } w \cdot f(x_i, y_i) - w \cdot f(x, \hat{y}) \geq L(y_i, \hat{y})$$

where $L(y_i, y)$ is a measure of the loss of using y instead of the correct labeling y_i, and $\hat{y}$ is a shorthand for $y_{w_{\text{old}}}(x_i)$. In case of a single constraint, this program has a closed-form solution. The most straightforward and the most commonly used loss function is the Hamming loss, which sets the loss of labeling y with respect to the gold labeling y_i as the number of training examples where the two labelings disagree. Since Hamming loss is not flexible enough for targeted training towards recall or precision, we use a number of task-specific loss functions (see Sections 3 and 5 for details). We implemented one-best MIRA and the corresponding loss functions in an in-house toolkit called Edlin. Edlin provides general machine learning architecture for linear models and a framework with implementations of popular learning algorithms including Naive Bayes, perceptron, maximum entropy, one-best MIRA, and conditional random fields (CRF) among others.

3 Trigger Word Tagging

The training and the development abstracts were first tokenized and split into sentences using maximum entropy models trained on the Genia[3] corpora. Subsequently, we trained several sequence taggers in order to identify the trigger words in text. All our experiments used the standard BIO encoding (Ramshaw and Marcus, 1995) with different feature sets and learning procedures. We focused on recall since it determines the upper bound on the performance of our final system. In our experiments, we found that simultaneously identifying trigger words and the event types they trigger yielded low recall; thus, we settled on identifying trigger words in text as one kind of entity, regardless of event types.

In our initial experiments, we used a CRF-based sequence tagger (Lafferty et al., 2001), which yielded R=43.51%. We further tried feature induction (McCallum, 2003) and second-order Markov assumptions for the CRF, achieving 44.72% and 49.64% recall, respectively.

[3] *http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/home/wiki.cgi*

Feature Set	R	P	F_1
Baseline (current word)	44.82	2.86	05.38
+ POS & char 3-gram	77.41	27.96	41.09
+ previous POS tag	79.77	29.32	42.88
+ lexicon (final tagger)	**80.44**	29.65	43.33

Table 1: Recall (R), precision (P), and F_1-measure for the trigger words tagger (in %s) on the development dataset for different feature sets using MIRA training with false negatives as a loss function.

Feature Sets
entity type of e_1 and e_2
words in e_1 and e_2
word bigrams in e_1 and e_2
POS of e_1 and e_2
words between e_1 and e_2
word bigrams between e_1 and e_2
POS between e_1 and e_2
distance between e_1 and e_2
distance between e_1 and e_2 in the dependency graph
steps in parse tree to get e_1 and e_2 in the same phrase
various combinations of the above features

Table 2: Our feature set for the MIRA classifier that predicts binary relations. Here e_1 and e_2 can be *proteins* and/or *trigger words*.

Subsequently, we settled on using MIRA so that we can trade-off precision for recall. In order to boost recall, we defined the loss function as the number of false negative trigger chunks. Thus, a larger loss update was made whenever the model failed to discover a trigger word, while discovering spurious trigger words was penalized less severely. We experimented with popular feature sets previously used for named entity (McCallum and Li, 2003) and gene (McDonald and Pereira, 2005) recognition including orthographic, part-of-speech (POS), shallow parsing and gazetteers. However, we found that only a small number of them was really helpful; a summary is presented in Table 1. In order to boost recall even further, we prepared a gazetteer of trigger chunks derived from the training data, and we extended it with the corresponding WordNet synsets; we thus achieved 80.44% recall for our final tagger.

4 Event Extraction

The input to our event extraction algorithm is a list of trigger words and a list of genes or gene prod-

ucts (e.g., proteins); the output is a set of relations as defined for Task 1. Our algorithm works in two stages. First, we generate events corresponding to relations between a trigger word and one or more proteins (*simple* events); then we generate events for relations between trigger words, proteins and simple events (*complex* events). The two stages differ only in the input data; thus, below we will describe our system for the first stage only.

For each sentence, we considered all pairs of entities (trigger words and proteins), and we used an unstructured classifier to determine the relationship for a given pair. These relationships encoded both the type of event (e.g., *binding*, *regulation*) and entities' roles in that event (e.g., *theme*, *cause*); there was also a special relationship for unrelated entities. We constructed labeled examples to train a MIRA classifier using the training data provided by the task organizers; n-ary relations were then reconstructed from classifier's predictions. The features we used are summarized in Table 2: they are over the words separating the two entities and their part-of-speech tags. We further used some simple features from syntactic phrases (OpenNLP[4] parser) and dependency parse trees (McDonald et al., 2005), extracted using parsers trained on Genia corpora.

After some initial experiments, we found that our features were not sufficiently rich to allow us to learn the relationships between proteins that are part of the same event: we achieved a very low recall of about 20%. Consequently, we focused on the relationships between a trigger word and a protein. Since the competition stipulated that each trigger could be associated with only one type of event, we first chose the event type for each trigger by selecting the protein-label pair with the highest score. We then fixed the event type for this trigger word, and we discarded all proteins for which our classifier assigned a different event type to the target trigger-protein pair. Finally, we added to our output list all binary relations where the role of the protein was *theme*.

For some event classes – *binding*, *regulation*, *positive regulation* and *negative regulation* – the output of the binary classifier was further transformed so that n-ary relations can be formed. However, the way we did this was somewhat ad-hoc. For *binding* events, we added a 3-ary relation between the trigger, the highest scoring protein, and the second highest scoring protein. For *regulation* events, we added a 3-ary relation between the trigger and every pair of proteins where one was a *theme* and the other one was a *cause*. This aggressive addition of potential matches slightly reduced the overall precision, but helped improve the recall for the final system.

[4]http://opennlp.sourceforge.net

Event Class	R	P	F_1
Localization	10.92	82.61	19.29
Binding	7.20	39.68	12.20
Gene expression	30.47	74.58	43.26
Transcription	10.95	39.47	17.14
Protein catabolism	28.57	57.14	38.10
Phosphorylation	34.07	86.79	48.94
Event Total	21.52	68.68	32.77
Regulation	1.37	26.67	2.61
Positive regulation	1.12	25.58	2.14
Negative regulation	0.26	100.00	0.53
Regulation Total	0.97	27.12	1.87
Overall	10.84	64.13	18.55

Table 3: Our official results: for an erroneous submission.

5 Results and Discussion

Unfortunately, we made an error when making our official submission, which resulted in low scores; Table 3 shows the results for that submission.

The rest of this section describes the results and the implementation for the system we *intended to submit*. All reported results are for exact span matches and were obtained using the online tool provided by the task organizers.

As stated in Section 4, we used a linear model trained using one-best MIRA with ten runs over the data for the event extraction system. We oversampled the unstructured training instances that corresponded to a relation so that they become roughly equal in number to those that do not correspond to a relation. Finally, we performed parameter averaging as described in (Freund and Schapire, 1999). These details turned out to be very important for the system performance.

Table 4 shows the results for three different loss functions that gave the best results in our experiments. In describing the loss functions, we define three different types of errors: (1) if the system correctly predicted that a relation should be present,

	0-1 Loss			High Recall			High Precision		
Event Class	R	P	$\mathbf{F_1}$	**R**	P	F_1	R	**P**	F_1
Localization	33.33	69.05	44.96	39.08	48.23	43.17	25.86	86.54	39.82
Binding	38.33	32.60	35.23	46.97	24.51	32.21	24.50	37.95	29.77
Gene expression	57.89	65.72	61.56	64.82	53.49	58.61	47.65	76.27	58.65
Transcription	30.66	33.87	32.18	33.58	22.12	26.67	21.17	47.54	29.29
Protein catabolism	42.86	85.71	57.14	42.86	60.00	50.00	42.86	85.71	57.14
Phosphorylation	75.56	77.86	76.69	77.78	65.22	70.95	52.59	82.56	64.25
Event total	49.64	54.60	52.00	55.98	41.55	47.70	37.93	65.83	48.13
Regulation	0.00	0.00	0.00	2.41	22.58	4.35	0.00	0.00	0.00
Positive regulation	1.73	30.91	3.28	5.29	25.24	8.75	0.20	28.57	0.40
Negative regulation	0.53	40.00	1.04	1.06	23.53	2.02	0.26	100.00	0.53
Regulation Total	1.15	30.16	2.21	3.81	24.80	6.61	0.18	37.50	0.36
Overall	24.45	53.54	**33.57**	**28.88**	39.71	33.44	18.32	**65.58**	28.64

Table 4: Results (in %s) for one-best MIRA with different loss functions.

but guessed the wrong type, we call this a *cross-labeling*; (2) a *false positive* occurs when the learner guessed some relation while there should have been none; (3) the reverse is a *false negative*. All loss functions we considered had a cross-labeling loss of 1. The *0-1 loss* also has a loss of 1 for false positives and false negatives. The *high-recall loss* function penalizes false positives with 0.1 and false negatives with 5. The *high-precision loss* function penalizes false negatives with 0.1 and false positives with 5. The values 0.1 and 5 were chosen on the development data, but were not optimized aggressively.

In conclusion, we have built three domain-independent event extraction systems based on the MIRA framework, each using a different loss function. Overall, they perform quite well and would have been ranked second on precision[5], and 6^{th} on recall, and 7^{th} on F_1-measure.

6 Future Work

After integrating domain knowledge, which should improve the recall for complex events and should boost the overall precision, we intend to transform the system output into RDF and add it to the PIKB repository. The required efforts discouraged us from building a middle ontology between the BioNLP and the PIKB data models, especially given the time limitations for the present task competition. However, we believe this is a promising direction, which we plan to pursue in future work.

[5]Our official submission is second on precision as well.

Acknowledgments

The work reported in this paper was partially supported by the EU FP7 - 215535 LarKC.

References

Koby Crammer. 2004. *Online Learning of Complex Categorial Problems*. Ph.D. thesis, Hebrew University of Jerusalem.

Yoav Freund and Robert E. Schapire. 1999. Large margin classification using the perceptron algorithm. In *Machine Learning*, pages 277–296.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*. Morgan Kaufmann.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of CoNLL*.

Andrew McCallum. 2003. Efficiently inducing features of conditional random fields. In *Proceedings of UAI*.

Ryan McDonald and Fernando Pereira. 2005. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*, (Suppl 1):S6(6).

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL*. ACL.

Lawrence Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2).

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*. ACL.