JMLR: Workshop and Conference Proceedings 1: 112-121 Machine Learning in Systems Biology

Event based text mining for integrated network construction

Yvan Saeys YVAN.SAEYS@PSB.UGENT.BE
Sofie Van Landeghem SOFIE.VANLANDEGHEM@PSB.UGENT.BE
Yves Van de Peer YVES.VANDEPEER@PSB.UGENT.BE

Department of Plant Systems Biology, VIB
Department of Molecular Genetics, University of Ghent
9052, Gent, Belgium

Editor: Sašo Džeroski, Pierre Geurts, and Juho Rousu

Abstract

The scientific literature is a rich and challenging data source for research in systems biology, providing numerous interactions between biological entities. Text mining techniques have been increasingly useful to extract such information from the literature in an automatic way, but up to now the main focus of text mining in the systems biology field has been restricted mostly to the discovery of protein-protein interactions.
Here, we take this approach one step further, and use machine learning techniques combined with text mining to extract a much wider variety of interactions between biological entities. Each particular interaction type gives rise to a separate network, represented as a graph, all of which can be subsequently combined to yield a so-called integrated network representation. This provides a much broader view on the biological system as a whole, which can then be used in further investigations to analyse specific properties of the network.

Keywords: Text mining, event extraction, integrated networks

1. Introduction

A wealth of biological information is currently recorded in scientific publications, which are easily accessible through online literature services like PubMed[1]. However, such resources are expanding exponentially and in order to keep up with the recent literature and retrieve relevant biological information, automated systems have become a time saving necessity.

Text mining methods are data mining techniques that focus on extracting relevant knowledge from these largely unstructured texts. Their use in systems biology started with simple, co-occurrence based methods that suggested relations between entities when they appeared in the same sentence (Ding et al., 2002), typically exhibiting high recall, but low precision (Hoffmand and Valencia, 2004). As high precision frameworks are often preferred in systems biology, especially when integrating different data sources, more elaborated techniques, either based on hand-crafted rules (Fundel et al., 2007) or machine

1. http://pubmed.gov

learning methods have been introduced. We will focus here on the latter techniques as they scale better to large datasets, and can be easily retrained when more data becomes available.

Up to now, the main focus of text mining techniques that rely on machine learning approaches has been the automatic extraction of protein-protein interactions, or the association of genes to certain diseases. A number of evaluation corpora have been built to assess the performance of techniques on the first of these tasks (Pyysalo et al., 2008; Van Landeghem et al., 2008a). Recently, the BioNLP'09 shared task was initiated as a community-wide effort to leverage the scope of text mining techniques to extract more complex events from text, in order to capture a wider variety of interactions and thus gain more knowledge from information encoded in the literature (Kim et al., 2009).

The main task in this challenge was to identify as good as possible 9 different types of bio-molecular events. For each event, the organizers provided a set of annotated PubMed abstracts, which could be used by the participants to train their models. Afterwards, a separate validation set was provided, allowing participants to evaluate their predictions, and finally an independent test set was provided to which all participants were evaluated.

In this work, we describe a machine learning approach that uses graph-based features from sentence representations to detect these different types of interactions, and subsequently uses them to construct an integrated network that contains all high-confidence predictions. The remainder of the manuscript is structured as follows. First, we elaborate on the methodology we used to convert these problems into a machine learning setting, outlining the general preprocessing of the documents, the applied machine learning techniques, and the final postprocessing to ensure a high-precision approach. Next, we present the results of this analysis: the evaluation of the whole framework on the BioNLP'09 evaluation and test set, and the construction of an integrated network using these predictions. We conclude by highlighting future perspectives and challenges that remain in this domain.

2. Methods

The core part of the BioNLP'09 challenge concerned the automatic detection and characterization of bio-molecular events from text. There are 9 distinct event types, six of which influence proteins directly, further referred to as 'Protein events', and three which describe 'Regulation events'. Five of the protein events are unary: Localization, Gene expression, Transcription, Protein catabolism and Phosphorylation. The sixth protein event, Binding, can be either related to one protein (e.g. protein-DNA binding), two proteins (e.g. protein-protein interaction) or more (e.g. a complex). The three types of Regulation events are the following: Regulation (unspecified), Positive regulation and Negative regulation. Each of them can be unary or binary. In the latter case, an extra argument specifying the cause of the regulation is added. Each argument of a Regulation event can be either a protein or any other event. This offers opportunities to detect different levels of interactions, and thus detect Regulation events in an iterative way.

The detection of Protein and Regulation events can now be stated as a set of binary classification problems, one for each event. A given potential occurrence of an event should then be scored by a classification model, which would either accept or reject the current

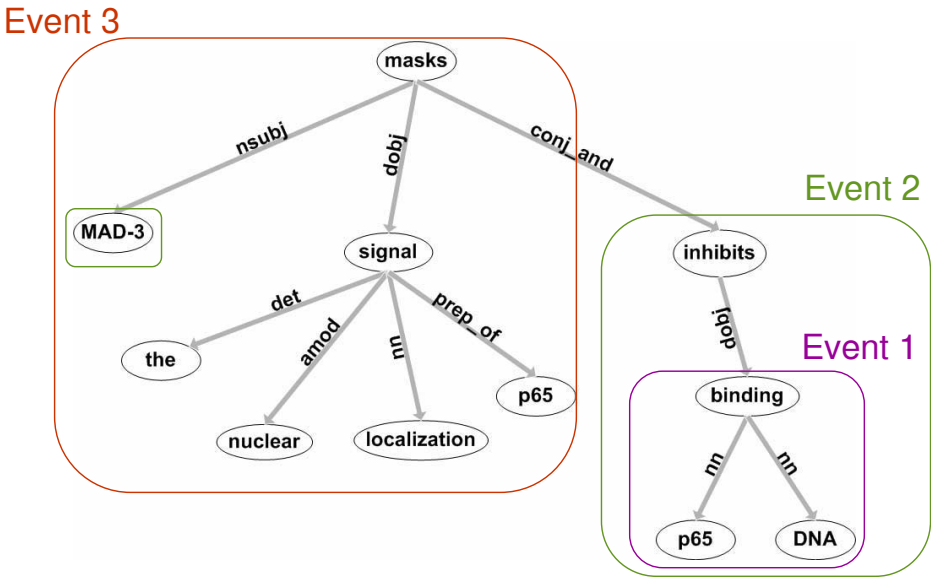

Figure 1: Example of a dependency graph for the sentence 'MAD-3 masks the nuclear localization signal of p65 and inhibits p65 DNA binding'. The three events represented in this sentence are indicated in the respective subgraphs.

example as being an instance of the particular event type. We will now go into more detail on how to transform the unstructured text data into a well defined classification task.

2.1 Data preprocessing

A challenging problem in text mining is to find an appropriate representation of the text, allowing machine learning techniques to make use of features that represent the key information to solve the task at hand. A few steps should be performed in order to transform the data into such a useful format.

In a first step, informative sentences containing biological entities are selected (information retrieval), and those key entities are identified and tagged in the sentence (named entity recognition). Subsequently, a deep syntactic parsing of each sentence was performed using the Stanford parser (de Marneffe et al., 2006), resulting in part-of-speech tags and dependency graphs. A dependency graph models the syntactic structure of a sentence, and is often used in many machine learning approaches as a structured data type to be used as input for the classification model (Zelenko et al., 2003; Kim et al., 2008).

Figure 1 shows an example of a dependency graph for the sentence 'MAD-3 masks the nuclear localization signal of p65 and inhibits p65 DNA binding'. This sentence contains three events to be detected by the system: 1) a Binding event (p65 DNA binding), 2) a Negative Regulation event (MAD-3 masks the nuclear localization signal of p65) and 3) a higher level Negative regulation event (MAD-3 inhibits p65 DNA binding), where one

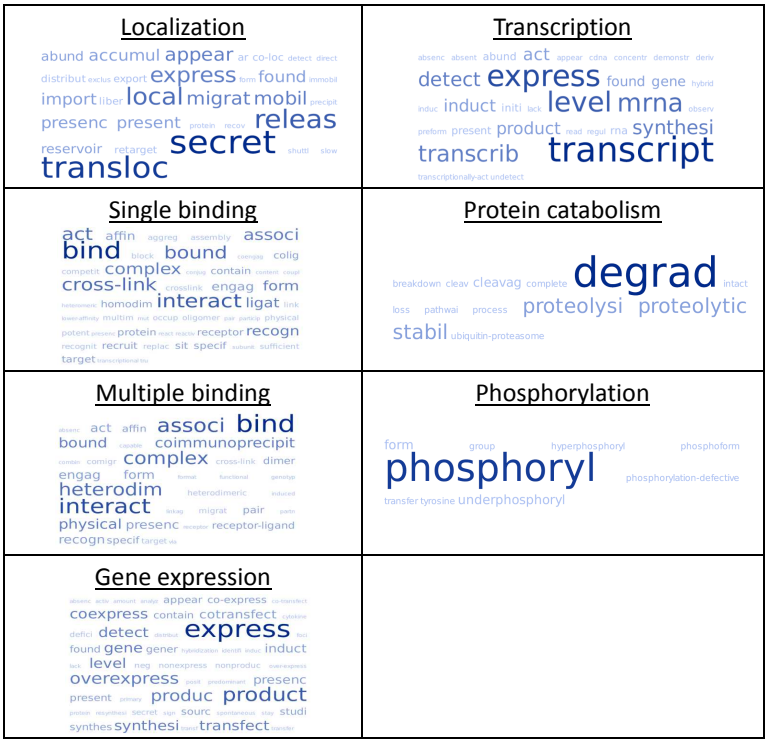

Table 1: Most important trigger words associoated to each event type.

of the arguments is a protein (MAD-3) and the other is an event in itself (p65 DNA binding).

To couple the words occurring in a sentence to a particular event, dictionaries of trigger words associated to each event were used (e.g. 'interaction' for Binding and 'secretion' for Localization). From the training data, we automatically compiled such dictionaries of triggers for each event type, applying the Porter stemming algorithm (Porter, 1980) to each trigger. This resulted in some entries in the dictionaries which were of limited use, such as 'through' for Binding, or 'are' for Localization. Such words are too general or too vague, and will lead to many negative and irrelevant instances. For this reason, we manually cleaned the dictionaries, only keeping specific triggers for each event type 1.

2.2 Model setup

To extract useful features from the dependency graph, we used a rich feature representation based on our earlier work on predicting protein-protein interactions (Van Landeghem et al., 2008b). The feature sets are a combination of information derived from the dependency tree (such as properties of the subgraph covering the event and lexical information of the

Event type	# Features	**# neg. inst.**	**# pos. inst.**	**% pos. inst.**
Localization	18 121	3415	249	7
Single binding	21 332	3548	522	13
Multiple binding	11 228	2180	185	8
Gene expression	31 332	5356	1542	22
Transcription	30 306	6930	489	7
Protein catabolism	1 883	175	96	35
Phosphorylation	2 185	163	153	48
Unspecified regulation (Unary)	27 915	6076	408	6
Positive regulation (Unary)	48 944	13834	1367	9
Negative regulation (Unary)	16 673	3233	489	13
Unspecified regulation (Binary)	4 239	778	81	9
Positive regulation (Binary)	19 468	5405	249	4
Negative regulation (Binary)	4 166	819	29	3

Table 2: Statistics of the training data set.

trigger words) and information concerning the occurrence of words in the subgraph. The following features were extracted:

- A bag-of-words (BOW) approach which looks at all the words that appear at a vertex of the subgraph. This automatically excludes uninformative words such as prepositions. Here we used stemmed trigrams (succesions of three words) as BOW features.
- Lexical and syntactic information of triggers (stemmed versions of each word, as well as the associated part-of-speech tag generated by the parser).
- Size of the subgraph.
- Length of the sub-sentence.
- Additional features for Regulation events, storing whether the arguments are proteins or events, and specifying the exact event type.
- Vertex walks, which consist of two vertices and their connecting edge. For these patterns, again lexical as well as syntactic information is kept. When using lexical information, protein names and triggers were blinded in order to extract more general patterns (e.g. 'trigger nsubj protx' which expresses that the given protein is the subject of a trigger).

The resulting datasets are inherently high-dimensional and very sparse. Table 2 shows the statistics of the training set for all event types. To deal well with these sparse, high-dimensional and class imbalanced datasets, we chose to use support vector machines (SVM) as the classification model (Boser et al., 1992). We used the LibSVM implementation of WEKA for our experiments, using the radial basis function (RBF) kernel as a default. As we were confronted with a separate validation and test set, only an internal 5-fold crossvalidation loop on the training data was used to optimize the C-parameter of the

SVM, and the classification performance on the validation and test sets were used to assess model performance.

Finally, a number of custom-made post-processing modules were applied to the resulting predictions, aiming to further reduce false positives and hence improve the precision of our method. These include removing the weakest predictions if multiple events were predicted for the same trigger word, as well as reducing the number of predictions based on overlapping trigger words.

2.3 Integrated network construction

We take a graph based approach to combine the predictions of the different Protein and Regulation events. Consider a set of interaction events $\{I_1, I_2, \cdots, I_N\}$ to integrate into a network. We can then associate to each of the events I_i a graph G_i, obtained using the predictions of the SVM model for event I_i. Note that there exists a heterogeneity in the graphs, as there might be multiple edges between two nodes in a graph (due to more than one prediction for a certain edge), and that some of the edges may be directed (e.g. A regulates B) while others may be undirected (e.g binding of C and D). Furthermore, all edges are weighted by the confidence of the associated prediction (see further).

A convenient representation for each graph G_i is its associated matrix $G_i(jk)$ where each entry in the matrix is a *set* of weighted connections between node j and node k. If there is no edge between node j and node k, then $G_i(jk) = \emptyset$. For undirected edges, the associated weight w_{jk} is represented both in $G_i(jk)$ and $G_i(kj)$, while for directed edges the weight is only added to the set representing the correct direction. This representation thus generalizes both directed and undirected information.

The weights on the edges are obtained by the classification model. For the SVM models, the distance to the hyperplane of each prediction is scaled between 0 and 1 such that the prediction threshold above which to decide on a positive prediction (this threshold varies per event) corresponds to a weight of 0.5.

It has to be noted that for some unary events, we may only know the effect, but not the causal node. In these cases, we introduce an artifical causal node for the effect node, which may be filled in later when more text is analysed. An integrated network can then be constructed by aggregating all matrices $G_i(jk)$ into a three-dimensional array $T(jkl)$ with dimensions $M \times M \times N$, where M is the cardinality of the union of all nodes in $G_i, i = 1 \cdots N$ and N is the number of events to integrate. The array entry $T(jkl)$ represents a connection from node j to node k for event type l. For visualisation purposes, we only keep all positive predictions, and discard all edges for which $T(jkl) < 0.5$.

3. Results

3.1 Predictive performance

To evaluate predictive performance, participants of the BioNLP'09 challenges could make use of a validation set to eventually fine-tune some parameters of their systems. However, performance could only be measured indirectly by submitting the predictions through a

Event type	Validation set			Test set		
	Recall	**Precision**	**F-score**	**Recall**	**Precision**	**F-score**
Localization	77.36	91.11	83.67	43.68	78.35	56.09
Binding	45.16	37.21	40.80	38.04	38.60	38.32
Gene expression	70.79	79.94	75.08	59.42	81.56	68.75
Transcription	60.98	75.76	67.57	39.42	60.67	47.79
Protein catabolism	80.95	89.47	85.00	64.29	60.00	62.07
Phosphorylation	68.09	88.89	77.11	56.30	89.41	69.09
Regulation	23.67	41.67	30.19	10.65	22.79	14.52
Positive regulation	21.56	38.00	27.51	17.19	32.19	22.41
Negative regulation	30.10	41.26	34.81	22.96	35.22	27.80

Table 3: Performance evaluation of all events for the validation and test datasets.

Team	Protein Events	Binding	Regulation	All
UTurku	70.21	44.41	40.11	51.95
JULIELab	68.38	41.20	34.60	46.66
ConcordU	61.76	27.20	35.43	44.62
UT+DBCLS	63.12	31.19	32.30	44.35
VIBGhent	**64.59**	**38.32**	**22.41**	**40.54**
UTokyo	55.96	41.10	20.09	36.88
UNSW	55.39	28.92	20.90	34.92
UZurich	53.66	33.75	19.89	34.78
ASU+HU+BU	56.82	27.49	09.01	32.09
Cam	51.79	18.14	15.79	30.80

Table 4: Performance comparison for the top ten performing teams. Numbers shown denote the F-measure for the three types of events (columns Protein, Binding, and Regulation), as well as the overall performance (column All).

web interface, which then returned the evaluation measures (recall, precision and F-score). This only allowed for a rough, manual fine-tuning of some of the systems parameters, as an automatic exploration of parameter settings using this web interface was not possible. In our case, we only fine-tuned for each event the prediction threshold above which to consider a prediction to be positive.

Similarly, the final results on the test set were also assessed in a blind way: participants could only upload their predictions for this set one time, and after the submission deadline all evaluations were returned to the participants. Table 3 shows the evaluation measures for our system on both the validation (using optimized thresholds) and test set.

As can be expected, performance on the test set is lower than on the validation set, the decrease in F-measure ranging from only about 2% for Binding events, to 27% in the case of Localization events. In general, we achieve a high precision for Protein events: almost all results achieve a precision of 60% or more. Another trend is the fact that predicting

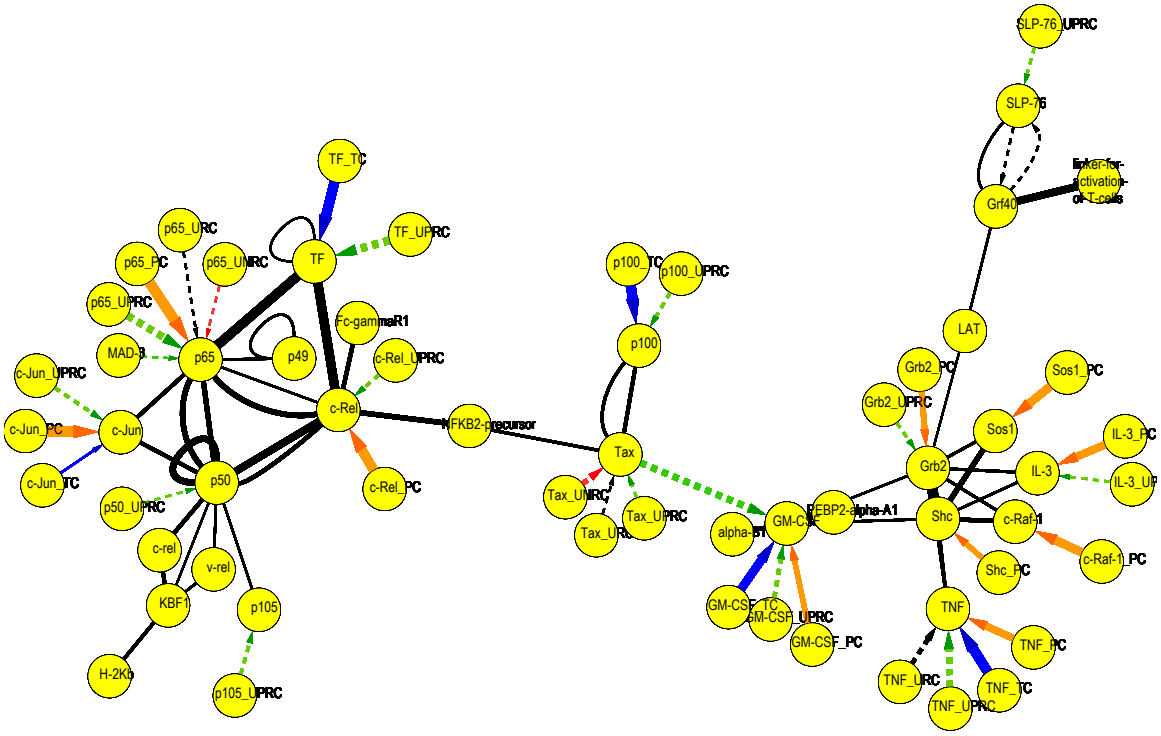

Figure 2: Visualization of a subgraph of the integrated network, constructed on the combined results of the validation and the test set.

Protein events achieves much higher performance than the prediction of Regulation events, a phenomenon that was observed by all participants in the challenge. This can be explained by the fact that the prediction of Regulation events largely depends on predicted Protein events (e.g. for higher level regulation events), thus causing false positives of predicted Protein events to cause even more false positive higher level regulation events.

To put these results into the context of the BioNLP'09 challenge, Table 4 compares the results of the ten best performing teams, out of 24 participating teams. Our team (VIBGhent) was ranked third for detecting Protein Events, fourth for detecting Binding Events, and fifth for detecting Regulation Events, resulting in an overall fifth ranking.

3.2 Constructing integrated networks

We created the multidimensional array $T(jkl)$ for a set of six events $\{I_1, I_2, I_3, I_4, I_5, I_6\} =$ {Positive regulation,
Negative regulation, Unspecified regulation, Binding, Transcription, Phosphorylation}. Figure 2 shows a visualization of a subgraph of the integrated network, where the edge thickness corresponds to the prediction confidence of the interaction, and colors display different types of interactions (black for Binding and unspecified Regulation events, orange for Phosphorylation, blue for Transcription and green/red for Positive/Negative Regulation events).

Furthermore, Regulation events are displayed by dashed lines, and Protein events by full lines.

In a subsequent stage, the array $T(jkl)$ can be used to infer new biological knowledge, such as indirect interactions and pathways. An example of an indirect interaction, derived from the network depicted in Figure 2 is the positive regulation of GM-CSF by Tax, which is in turn negatively regulation by Tax_UNRC, which suggests an indirect regulation of GM-CSF by Tax_UNRC.

4. Conclusions and future work

In this work we presented a text mining approach that extracts various types of interactions from scientific literature. This information was used in a second stage to construct integrated networks, using the strength of the predictions as confidence weights for the connections in the network. As the application of text mining techniques for such problems is still in its childhood, improving the predictive performance of these techniques will remain a key challenge, as well as recognizing more adequately the specific type of interaction (e.g. protein-protein, protein-DNA, RNA-protein). Furthermore, we already performed some preliminary work on detecting speculation and negation of biological events, which will be useful to detect modes of (un)certainty about certain facts stated.
From a data integration point of view, we aim to combine the results obtained by text mining with interactions identified by other data sources (either experimentally verified or predicted) in order to increase the robustness of the networks.

References

B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

M.C. de Marneffe, B. Maccartney, and C. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC-06*, pages 449–454, 2006.

J. Ding, D. Berleant, D. Nettleton, and E. Wurtele. Mining medline: abstracts, sentences, or phrases? In *Proceedings of PSB'02*, pages 326–337, 2002.

K. Fundel, R. Küffner, and R. Zimmer. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3): 365–371, 2007. ISSN 1367-4803.

R. Hoffmand and A. Valencia. A gene network for navigating the literature. *Nature Genetics*, 36(7):664, 2004.

J.D. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii. Overview of bionlp'09 shared task on event extraction. In *BioNLP '09: Proceedings of the Workshop on BioNLP*, pages 1–9, Morristown, NJ, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-44-2.

J.D. Kim, T. Ohta, and J. Tsujii. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9 (1), 2008. URL `http://dx.doi.org/10.1186/1471-2105-9-10`.

M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980. URL `http://portal.acm.org/citation.cfm?id=275705`.

S. Pyysalo, A. Airola, J. Heimonen, J. Bjrne, F. Ginter, and T. Salakoski. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, 9(Suppl. 3):S6, 2008.

S. Van Landeghem, Y. Saeys, B. De Baets, and Y. Van de Peer. Benchmarking machine learning techniques for the extraction of protein-protein interactions from text. In *Proceedings of the 18th Belgian Dutch Machine Learning Conference (Benelearn'08)*, pages 79–80, 2008a.

S. Van Landeghem, Y. Saeys, B. De Baets, and Y. Van de Peer. Extracting protein-protein interactions from text using rich feature vectors and feature selection. In *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM)*. Turku Centre for Computer Sciences (TUCS), 2008b. URL `http://hdl.handle.net/1854/LU-538895`.

D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *JMLR*, 3(Feb):1083–1106, 2003.