A Hybrid Approach to Recognising Discourse Causality in the Biomedical Domain

Claudiu Mihăilă and Sophia Ananiadou
The National Centre for Text Mining
School of Computer Science, The University of Manchester,
131 Princess Street, Manchester, M17 DN, UK
Email: {claudiu.mihaila, sophia.ananiadou}@manchester.ac.uk

***Abstract*—Whilst current domain-specific information extraction systems represent an important resource for biomedical researchers, the increasing amount of knowledge published daily is still overwhelming them. As such, automatic discourse causality recognition can further improve the search for relevant information by suggesting possible causal connections. We describe here an approach to the automatic recognition of discourse causality in the biomedical domain using a combination of machine learning and rules. We test and evaluate our system on BioCause, a corpus containing gold standard annotations of causal relations. The best performance in identifying triggers is achieved by CRFs with 79.35% F-score. We then locate the arguments using naïve syntactic rules, achieving F-scores of around 90% in most cases. Determining which argument plays which role is performed by a group of machine learners with an F-score of 84.35%.**

***Keywords*-discourse analysis; biomedical causality;**

I. Introduction

The increasing amount of published biomedical knowledge in the form of research articles has driven significant recent advancements in biomedical text mining, including named entity recognition, event extraction and coreference resolution. Using biomedical text mining technology, text can now be enriched via the addition of semantic metadata and thus can support tasks such as analysing molecular pathways and semantic searching.

However, a large part of the undertaken research considers only particular facts that are expressed in at most one sentence if not one clause. For complex tasks, such as question answering and automatic summarisation, the extraction of information spanning several sentences is requried, together with the recognition of relations that exist across sentence boundaries. These relations, that connect facts in a logical manner, help readers infer deeper, more complex knowledge about the facts mentioned in the discourse. They can be either explicit or implicit, depending whether or not they are expressed in text using overt *discourse connectives* (also known as *triggers*). Take, for instance, the case in example (1), where the trigger *Therefore* signals a justification between the two sentences.

(1) In the case of PmrB, a normal response to mild acid pH requires not only a periplasmic histidine but also several glutamic acid residues. *Therefore*, regulation of PmrB activity may involve protonation of one or more of these amino acids.

Thus, by identifying these types of causal relations, search engines become able to automatically discover novel relations between biomedical entities, processes and events or between experimental evidence and associated conclusions. If the mechanism is applied to a large collection of articles, there are high chances of discovering facts that might have been overlooked by humans. However, phrases acting as causal triggers in certain contexts may not denote causality in all cases. In examples (2) and (3), the same trigger, *result in*, can have either a causal or a non-causal role, respectively, in the understanding of biomedical experts.

(2) Inactivation of crdA has been shown to *result in* delayed M. xanthus multicellular development.

(3) However, continued incubation did not *result in* increased DeltamifR biofilm growth (biomass, thickness) or microcolony formation typically seen in wild type biofilms at the maturation-2 stage (post 72 hr of growth).

In this paper, we explore a hybrid approach to the automatic identification of triggers that actually denote causality, their arguments and their arguments' roles.

II. Related Work

Although prior studies in discourse causality extraction are relatively sparse, a large amount of work has been dedicated to generic discourse parsing and discourse relation identification in the general domain, where researchers have developed end-to-end discourse parsers. Most work is based on the Penn Discourse Treebank (PDTB) [1], a corpus of lexically-grounded annotations of discourse relations. Whilst there are many successful attempts in the direction of automatically classifying triggers, argument identification has been explored to a more limited extent.

For instance, some researchers have disambiguated discourse triggers from non-triggers without determining the discourse relation [2]. Using almost only syntactic features related to the trigger, they achieve an F-score of around 95%.

978-1-4799-1310-7/13/$31.00 ©2013 IEEE

Another two approaches consider the syntactic constituency and dependency structure of the trigger's context [3] and combining ceartain aspects of the surface level and syntactic feature sets of these respective works [4].

Regarding the arguments, researchers have undertaken the restricted task of identifying their heads instead of their actual spans [5]. Others limit their detection to identifying only the sentence in which the argument is located [6], whilst complete arguments with boundaries have been identified only for a restricted set of discourse relations [7]. Considering triggers as input and thus focussing only on arguments, some studies combine syntactic and semantic features to identify arguments as a sequence labelling task [8]. A complete discourse parser has been built by introducing new features in all steps of discourse parsing [9], which improves on the previous state-of-the-art.

Until now, comparatively little work has been carried out on causal discourse relations in the biomedical domain, although it has been proven that discourse triggers differ between the biomedical and general domains by training a model on the BioDRB and testing it against PDTB and vice-versa [10], [4]. The equivalent of the PDTB for the biomedical domain is the BioDRB corpus [11]. A larger corpus in terms of causal relations is BioCause [12], containing manually annotated causal discourse relations.

Using BioDRB as data, some researchers have explored the identification of discourse connectives [10]. However, they do not distinguish between the types of discourse relations and, based on orthographic features, they obtain the best F-score of 75.7% using CRFs.

Unlike previous work, we develop a methodology for an end-to-end discourse parser focussed on causality as expressed in the biomedical domain, and evaluate its performance on BioCause.

III. Methodology

In this section, we describe our data and the features used for causality identification, and provide an overview of the processing pipeline. We also explain our evaluation methodology.

A. Data

The data for the experiments comes from the BioCause corpus, a collection of 19 journal articles on infectious diseases, manually annotated with 850 causal relationships. Two types of spans of text are marked in the text, namely causal triggers and causal arguments. Each causal relation is composed of three text-bound annotations: a trigger, a cause or evidence argument and an effect argument. A small number of causal relations have implicit triggers, so these are excluded from the current research. Fig. 1 shows an example of discourse causality from BioCause, marking the causal trigger and the two arguments with their respective relation. Named entities are also marked in this example.

Require: a text T
Ensure: discourse causal relations in T
Identify all causal triggers in T
for all trigger t **do**
  Label t as SS or DS
  if t is SS **then** {arguments in same sentence}
    Split sentence in clauses
    Label the immediate right clause of t as DepArg
    Label the rest of the sentence as IndArg
  else {arguments in different sentences}
    Label sentence of t as DepArg
    Identify IndArg around the sentence of t
  end if
  Identify relation direction
end for

Figure 2. Pseudocode for identifying causal relations in the BioCause.

BioCause contains 381 unique explicit triggers in the corpus, each being used, on average, only 2.10 times. The number decreases to 347 unique triggers when they are lemmatised, corresponding to an average usage of 2.30 times per trigger. Both counts show the diversity of causal phrases that are used in the biomedical domain. The order of the arguments does not vary significantly, with more than 85% occurring in the form of cause-trigger-effect (C-T-E).

B. Pipeline

The pseudocode for the causality recognition pipeline is shown in Figure 2. Similar to the annotation mechanism used by the experts who produced the BioCause corpus, we have split the recognition of causality into three major steps. First, the annotators were given just the raw text T, which was then analysed to find causal triggers. We modelled trigger detection both as a classification task, using Support Vector Machines (LibSVM [13]) and Random Forests (Weka [14]), and as a sequence labelling task, using CRF [15] (CRFSuite). Second, when a causal trigger was found, the annotators decided on whether its two arguments are in the same sentence (SS) or different sentences (DS). In the former case, the clause syntactically depending on the trigger becomes the dependent argument (DepArg), whilst the rest of the sentence represents the independent argument (IndArg). In the latter case, the sentence containing the trigger becomes the dependent argument, whilst the independent argument is identified as one of the sentences around the trigger. We used a machine learner to distinguish between intra- and inter-sentential relation and rules to detect the argument spans. Finally, in the third step, after both arguments are located, the annotator classifies the direction of the relation, that is which argument plays which of the semantic roles of cause and effect. We trained several models to assign roles to the previously detected arguments.

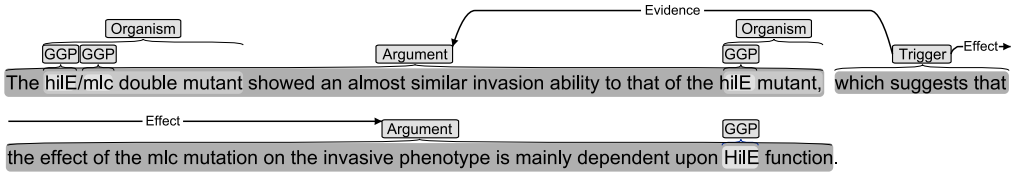


Figure 1. Causal relation in the BioCause.

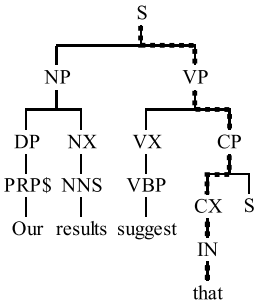


Figure 3. Partial lexical parse tree of a sentence starting with a causal trigger.

C. Features

Three types of features have been employed in the development of this causality model, i.e., lexical, syntactic and semantic. They are categorised and described below.

1) Lexical features: The lexical features are built from the actual tokens present in text, identified by the GENIA tagger [16] trained on MEDLINE. Features include the token's surface expression, its lemma and the three immediately neighbouring tokens. This decision is based on two observations. Firstly, in the case of tokens to the left, most triggers are found either at the beginning of the sentence (311 instances) or are preceded by a comma (238 instances). These two left contexts represent 69% of all triggers. Secondly, for the tokens to the right, almost 45% of triggers are followed by a determiner, such as *the*, *a* or *an*, (281 instances) or a comma (71 instances).

2) Syntactic features: The syntax, dependency and predicate argument structure are produced by Enju [17]. Figure 3 depicts a partial lexical parse tree of a sentence which starts with a causal trigger, namely *Our results suggest that*, from which several types of features have been generated. Features include the part-of-speech and syntactic category of a token, since either many triggers are lexicalised as an adverb or conjunction, or are part of a verb phrase. For the same reason, the syntactical category path from the root of the lexical parse tree to the token is included. The path also encodes, for each parent constituent, the position of the token in its subtree, i.e. beginning (*B*), inside (*I*) or end (*E*); if the token is the only leaf node of the constituent, this is marked differently, using a *C*. Thus, the path of *that*, highlighted in the figure, is *I-S/I-VP/B-CP/C-CX*.

Secondly, for each token, we extracted the predicate argument structure and checked whether a relation exists between the token and the previous and following tokens. The values for this feature represent the argument number as allocated by Enju.

Thirdly, the ancestors of each token to the third degree are instantiated as three different features. In the case that such ancestors do not exist (i.e., the root of the lexical parse tree is less than three nodes away), a "none" value is given. For instance, the token *that* in Figure 3 has as its first three ancestors the constituents marked with *CX*, *CP* and *VP*.

Finally, the lowest common ancestor in the lexical parse tree between the current token and its left neighbour has been included. In the example, the lowest common ancestor for *that* and *suggest* is *VP*. These last two feature types have been produced on the observation that the lowest common ancestor for all tokens in a causal trigger is S or VP in over 70% of instances. Furthermore, the percentage of cases of triggers with V or ADV as lowest common ancestor is almost 9% in each case. Also, the average distance to the lowest common ancestor is 3.

3) Semantic features: We have exploited several semantic knowledge sources to identify causal triggers more accurately, as a mapping to concepts and named entities acts as a back-off smoothing, thus increasing performance.

One semantic knowledge source is the BioCause corpus itself. All documents annotated for causality in BioCause had been previously manually annotated with biomedical named entity and event information. We therefore leverage this existing information to add another semantic layer to the model, thus eliminating error propagation from automatic named entity recognition and event extraction.

Second, tokens have been linked to the Unified Medical Language System (UMLS) [18] semantic types. Thus, we included a feature to say whether a token is part of a UMLS type and another for its semantic type if the previous is true.

Third, the hypernym of every token in the text, extracted from WordNet [19], has been included as a feature. Only the first sense of every token has been considered, as no sense disambiguation technique has been employed.

IV. Results and discussion

Several models have been developed and 10-fold cross-evaluated to examine the complexity of the task.

Table I
PERFORMANCE OF VARIOUS CLASSIFIERS IN IDENTIFYING CAUSAL CONNECTIVES.

Classifier	P	R	F_1
Dict	0.08	1.00	0.15
Depend	0.08	0.77	0.14
Dict+Depend	0.14	0.75	0.24
CRF	0.89	0.74	0.79
SVM	0.88	0.61	0.70
RandFor	0.78	0.67	0.72

A. Trigger identification

Table I shows the performance of baseline systems and other classifiers. It should be noted that the training and testing datasets are highly skewed, with a ratio of positive examples to negative examples of approximately 1:52.

The first baseline is a dictionary-based heuristic, named *Dict*, whose precision is very low, 8.36%, which leads to an F-score of 15.43%, considering that the recall is 100%. This is mainly due to words and/or phrases which are rarely used as causal triggers, such as *and*, *by* and *that*.

Based on the lowest common ancestor for all tokens in a causal trigger, we built a baseline system that checks all constituent nodes in the lexical parse tree for the S, V, VP and ADV tags and marks them as causal triggers. The name of this system is *Depend*. It obtains a slightly lower precision than *Dict* and performs worse in terms of recall. The F-score is 13.68%, largely due to the high number of intermediate nodes in the lexical parse tree that have VP as their category.

The third baseline is a combination of *Dict* and *Depend*: we consider only constituents that have the necessary category (S, V, VP or ADV) and include a trigger from the dictionary. Although the recall decreases slightly, the precision increases to almost twice that of both *Dict* and *Depend*, producing a much better F-score of 24.03%.

In the case of CRFs, the best performance in terms of F-score is obtained when combining all three types of features. The best precision is obtained by using the syntactic features only, reaching over 92%, almost 3% higher than when all three feature types are used.

For SVMs, we have experimented with two kernels, namely second degree polynomial and radial basis function kernels. For each of these two kernels, we have evaluated various combinations of parameter values for cost and weight. Both these kernels achieved similar results, indicating that the feature space is not linearly separable and that the problem is highly complex. Regarding the effect of feature types on the performance of SVMs, the best performance is obtained when combining the lexical and semantic feature types (69.85% F-score). The combination of all features produces the best precision, whilst the best recall is obtained by combining lexical and semantic features.

The best performance of RFs is obtained when combining lexical and semantic features. Because causal triggers cannot be mapped to concepts in the named entity and UMLS annotations, the RF classifier can easily produce rules to distinguish triggers from non-triggers. As such, the use of semantic features alone produce a very good precision of 84.34%. Also, when semantics is added, the precision increases by 0.5% in the case of lexical features and 3.5% in the case of syntactic features. However, the recall of semantic features alone is the lowest. The best recall is obtained when using only lexical features.

As we expected, most errors arise from sequences of tokens which are used infrequently non-causally. Figure 4 shows the usage of annotated triggers as having causal and non-causal meaning in black and grey, respectively. The order of triggers in the two charts is preserved. Figure 4a depicts the actual number of instances for each trigger, whilst Figure 4b is based on the ratio of causal:non-causal instances for each trigger. A logarithmic scale is used in Figure 4a for visibility purposes, as there are many small values and very few large ones. By analysing both figures simultaneously, it can be noticed that there exists a large number of triggers which seldom occur, but which are exclusively causal in meaning. More than 200 triggers, to the left of both charts, occur less than 20 times each, but they are 100% causal.

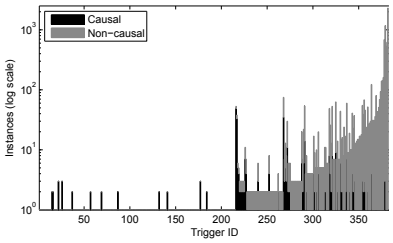
On the other hand, there are 107 trigger types whose number of false positives (FP) is higher than the number of true positives (TP). In fact, 64 trigger types occur only once as a causal instance, whilst the average number of FPs for these types is 14.25. One such example is *and*, for which the number of non-causal instances (2305) is much greater than that of causal instances (1). Other examples of trigger types more commonly used as causal triggers, are *suggesting* (9 TP, 54 FP), *indicating* (8 TP, 41 FP) and *resulting in* (6 TP, 14 FP). For instance, example (4) contains two mentions of *indicating*, but neither of them implies causality.

(4) Buffer treated control cells showed intense green staining with syto9 (*indicating* viability) and a lack of PI staining (*indicating* no dead/dying cells or DNA release).

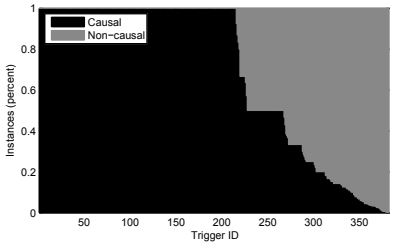
B. Argument identification

The process of identifying the two arguments of the causal trigger is divided into two steps. First, a classifier is built in order to determine whether the two arguments are located in the same sentence or not, based on the trigger. Second, based on the result of the previous step, two spans representing the arguments are located around the trigger, either in the same sentence or neighbouring sentences.

The classification of the trigger into SS or DS is performed by machine learning algorithms. We employ the trigger's surface form, base form, PoS and syntactic category, its position in the sentence and whether it contains the sentence's main verb. We have experimented with different types of algorithms implemented in Weka, ranging from



(a) Number of causal/non-causal instances for each annotated trigger



(b) Percentage of causal/non-causal usage for each annotated trigger

Figure 4. Usage of annotated triggers as having causal and non-causal meaning.

Table II
PERFORMANCE OF VARIOUS ALGORITHMS IN CLASSIFYING TRIGGERS AS SS OR DS.

Classifier	P	R	F_1
Näive Bayes	91.85%	91.90%	91.87%
SVM	92.75%	92.55%	92.65%
JRip	93.20%	92.65%	92.92%
J48	93.00%	92.80%	92.90%
RandFor	92.70%	92.80%	92.75%
Vote	94.75%	94.60%	94.65%

Table III
PERFORMANCE IN IDENTIFYING DEPENDENT (DA) AND INDEPENDENT (IA) ARGUMENT SPANS.

Argument-Case	P	R	F_1
DA-SS	74.36%	100%	85.29%
IA-SS	82.75%	96.53%	89.11%
DA-DS	84.64%	98.83%	91.18%
IA-DS	54.98%	60.58%	57.64%

simple probabilistic classifiers (Naïve Bayes) to decision trees (J48 and RF), rules (JRip) and support vector machines (SVM). The Vote meta-classifier is configured to consider the five previous classifiers, using an Average of Probabilities combination rule. Table II shows the macro-averaged performance of the employed classifiers. As can be seen, the performances are very similar in between all classifiers, their F-score ranging within just over 1% of 92%. Furthermore, the Vote meta-classifier improves the results only slightly, by 1.67% over JRip, which leads us to the conclusion that all classifiers make relatively the same decisions.

In the case of same-sentence arguments, we employ a naïve rule which splits the sentence into two segments, each on either side of the trigger. The shortest segment that is contained within an S or S-REL constituent and immediately follows the trigger is marked as the dependent argument, whilst the segment preceding the trigger is marked as the independent argument. The performance of this simple rule is impressive, reaching values between 85% and 90%, as can be noticed from Table III. More specifically, the evaluation result for dependent argument identification in the case of SS is F-score 85.29%. The recall reaches 100%, as all words after the token are marked as part of this argument. However, the precision only gets to almost 75%, showing that a better selection needs to be implemented to identify tokens that not part of the argument. On the other hand, the independent argument identification in the case of SS reaches a higher F-score of 89.11%.

Table IV
PERFORMANCE OF VARIOUS CLASSIFIERS IN IDENTIFYING CAUSAL DIRECTION.

Classifier	P	R	F_1
Näive Bayes	69.85%	83.80%	73.40%
SVM	81.70%	79.90%	80.80%
JRip	81.60%	80.35%	80.95%
J48	83.40%	79.15%	81.10%
RandFor	83.70%	72.55%	76.60%
Vote	85.25%	83.55%	84.35%

In case the two arguments of the causal trigger are classified as being in distinct sentences, we mark the entire sentence containing the trigger as the dependent argument. Thus, the dependent argument is marked as starting from the end of trigger to the end of the sentence. This leads to an F-score of 91.18%, with almost 99% recall and 85% precision. The independent argument is marked as the preceding sentence to that containing the trigger. This results in an F-score of 57.64%, much lower than those of other arguments. The two main reasons for a lower score are the syntactic independency and the large search space. Practically any sentence preceding or following the trigger sentence can play the role of the independent argument. The only possibility to improve the accuracy of identifying it is by employing deep semantics and other discourse features.

C. Relation direction identification

The final step in the causality recognition pipeline is to detect which argument plays which semantic role. Each of the previously identified arguments must be assigned one of the two possible roles, Cause and Effect. For this task, we

have explored a machine learning approach to detect whether a causal relation is of the form C-T-E or E-T-C. The other three possibilities existing in BioCause have been excluded from the classification, as their number is insufficient for training purposes. We have experimented with multiple algorithms, ranging from simple probabilistic classifiers (e.g., Naïve Bayes) to trees (e.g., J48 and Random Forests), rules (JRip) and support vector machines (SVM). The Vote meta-classifier is configured to consider the five previous classifiers, using an Average of Probabilities combination rule. All of the mentioned algorithms are used as implemented in Weka. Macro-averaged results are provided in Table IV. One aspect that has to be taken into consideration is the skewed data. The E-T-C to C-T-E ratio is 1:7.54. Under these circumstances, the best classifier, Vote, reaches an F-score of 96.40% for C-T-E and of 72.30% for E-T-C, resulting in a macro-average F-score of 84.35%. The most useful features in this classification, according to InfoGain and ChiSquare attribute evaluators, have proven to be the actual trigger, its lemmatised form, part-of-speech, syntactic category, its neighbours, the presence of the words *by*, *due* and pronouns, and the voice of the verb.

V. Conclusion

We have described our three-step approach to automatically recognise causal discourse relations in biomedical scientific text. First, we have explored various algorithms that automatically learn to distinguish triggers from non-triggers and we have evaluated the impact of multiple lexical, syntactic and semantic features. The best performing classifier is CRF-based and combines all features types to obtain an F-score of 79.35%. Second, we successfully identified three cases of arguments of causal triggers using shallow syntactic features, reaching F-scores of around 90%. The exception is when an argument is located in a different sentence, for whose identification deeper semantic knowledge is needed. The third step is related to labelling each argument identified above with Cause or Effect. We have trained several learners only on features of the trigger, the best of which is the Vote meta-classifier with an F-score of 84.35%.

More semantic information needs to be included in the second and third steps, which could improve the location of causal arguments, especially when they are not syntactically bound to the trigger. Furthermore, the addition semantics could more easily disambiguate between cause and effect.

Acknowledgement

This work was partially funded by the Engineering and Physical Sciences Research Council [grant number EP/P505631/1]; Medical Research Council; Europe PubMed Central Funders (led by Wellcome Trust).

References

[1] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber, "The Penn Discourse TreeBank 2.0," in *In Proceedings of LREC 2008*, 2008, pp. 2961–2968.

[2] E. Pitler and A. Nenkova, "Using syntax to disambiguate explicit discourse connectives in text," in *ACL/AFNLP (Short Papers)*, 2009, pp. 13–16.

[3] B. Wellner, "Sequence models and ranking methods for discourse parsing," Ph.D. dissertation, Brandeis University, 2009.

[4] S. Ibn Faiz and R. E. Mercer, "Identifying explicit discourse connectives in text," in *Advances in Artificial Intelligence*. Springer Berlin Heidelberg, 2013, vol. 7884, pp. 64–76.

[5] B. Wellner and J. Pustejovsky, "Automatically identifying the arguments of discourse connectives," in *Proceedings of EMNLP 2007*. ACL, 2007, pp. 92–101.

[6] R. Prasad, A. Joshi, and B. Webber, "Exploiting scope for shallow discourse parsing," in *Proceedings of LREC 2010*. Valletta: ELRA, 2010.

[7] N. Dines, A. Lee, E. Miltsakaki, R. Prasad, A. Joshi, and B. Webber, "Attribution and the (non-) alignment of syntactic and discourse arguments of connectives," in *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*. ACL, 2005, pp. 29–36.

[8] S. Ghosh, R. Johansson, G. Riccardi, and S. Tonelli, "Shallow discourse parsing with conditional random fields," in *Proceedings of IJCNLP 2011*, 2011, pp. 1071–1079.

[9] Z. Lin, H. T. Ng, and M.-Y. Kan, "A pdtb-styled end-to-end discourse parser," *Natural Language Engineering*, vol. FirstView, pp. 1–34, 10 2012.

[10] P. B. Ramesh, R. Prasad, T. Miller, B. Harrington, and H. Yu, "Automatic discourse connective detection in biomedical text," *JAMIA*, vol. 19, no. 5, pp. 800–808, 2012.

[11] R. Prasad, S. McRoy, N. Frid, A. Joshi, and H. Yu, "The biomedical discourse relation bank," *BMC Bioinformatics*, vol. 12, p. 188, 2011.

[12] C. Mihăilă, T. Ohta, S. Pyysalo, and S. Ananiadou, "BioCause: Annotating and analysing causality in the biomedical domain," *BMC Bioinformatics*, vol. 14, p. 2, Jan. 2013.

[13] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, 2011.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, 2009.

[15] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of ICML 2001*. San Francisco: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289.

[16] Y. Tsuruoka, Y. Tateishi, J.-D. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii, "Developing a robust part-of-speech tagger for biomedical text," in *Advances in Informatics - 10th Panhellenic Conference on Informatics*. Springer-Verlag, 2005, vol. 3746, pp. 382–392.

[17] Y. Miyao and J. Tsujii, "Feature forest models for probabilistic HPSG parsing," *Comp Ling*, vol. 34, no. 1, p. 3580, 2008.

[18] O. Bodenreider, "The unified medical language system (UMLS): integrating biomedical terminology," *Nucleic Acids Research*, vol. 32, no. suppl 1, pp. D267–D270, 2004.

[19] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.