Procedia Computer Science
Volume 53, 2015, Pages 274–283
2015 INNS Conference on Big Data

# Extracting Clinical Relations in Electronic Health Records Using Enriched Parse Trees

Jisung Kim[1], Yoonsuck Choe[2], and Klaus Mueller[3]

[1] Department of Computer Science, SUNY Korea, Incheon, South Korea
jskim@sunykorea.ac.kr
[2] Department of Computer Science and Engineering, Texas A&M University, College Station, TX, U.S.A.
choe@cs.tamu.edu
[3] Department of Computer Science, SUNY Korea, Incheon, South Korea.
mueller@cs.stonybrook.edu

---

**Abstract**

Integrating semantic features into parse trees is an active research topic in open-domain natural language processing (NLP). We study six different parse tree structures enriched with various semantic features for determining entity relations in clinical notes using a tree kernel-based relation extraction system. We used the relation extraction task definition and the dataset from the popular 2010 i2b2/VA challenge for our evaluation. We found that the parse tree structure enriched with entity type suffixes resulted in the highest F1 score of 0.7725 and was the fastest. In terms of reducing the number of feature vectors in trained models, the entity type feature was most effective among the semantic features while adding semantic feature node was better than adding feature suffixes to the labels. Our study demonstrates that parse tree enhancements with semantic features are effective for clinical relation extraction.

*Keywords:* relation extraction, clinical text, natural language processing, support vector machine, convolution tree kernel

---

## 1 Introduction

According to the Office of the National Coordinator for Health Information Technology (ONC), the US has invested more than 30 billion on incentive payments to hospitals and physicians to adopt EHRs (electronic health records) with more than 90% of the hospitals now attesting to most functionalities in meaningful use stage 1. For example, the number of Mayo Clinic's clinical notes accrued between 2000 and 2010 exceeds fifty one million [15]. The clinical records contained in these EHRs hold invaluable information for clinical decision support, effective cohort identification for clinical trials, adverse effect detection, as well as numerous other applications in healthcare and biomedical research (Fig. 1). However, to reap the benefits of such

Selection and peer-review under responsibility of the Scientific Programme Committee of INNS-BigData2015

doi:10.1016/j.procs.2015.07.304

big data in healthcare, the raw data must be properly converted to computable knowledge that can support good decisions and actions for the triple aims of (1) improved quality of care, (2) reduced costs, while (3) increasing access to care. In particular, information extraction (e.g., extracting entities and the relations between them) is critical to Big Data Analytics in the health domain. Information extraction is a key step to converting the numerous free text clinical notes into computable knowledge. For example, if we extracted all "a treatment worsens a medical problem" relations from discharge summaries and stored in a database, a physician can avoid such treatment while planning care for a patient who experienced such event previously, or if such negative effect is prevalent in a specific group, for example, people who are taking a certain combination of medications, this can be interpreted as a sign of side effect of medications not discovered in medical trials of those medications.

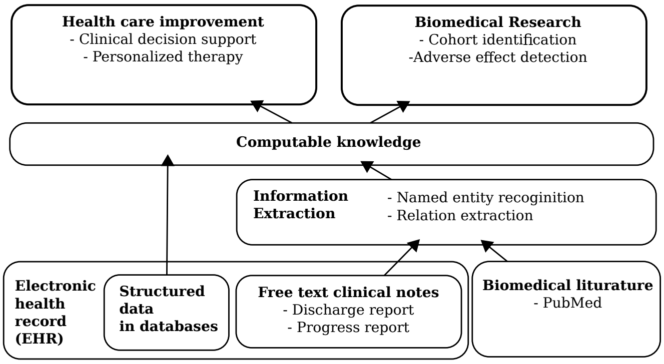

Figure 1: Role of information extraction in biomedical domain

In this paper, we focus on the problem of relation extraction that finds predefined relation types between two entities in free text clinical notes. Supervised machine learning methods have been prevalent in the recent clinical natural language processing (NLP) research. With supervised learning methods, investigations have been focused on creating more and better annotated data, designing better features, and improving the clinical linguistic model. Since access to raw data and expertise to annotate data are limited, many existing methods have focused on developing better features and classification strategies. However, a complex feature set would not be a proper way to approach clinical notes in a big data situation, because the cost and time of extracting features from raw data are not trivial, and errors in extracted features propagate through the processing pipeline and can harm the relation extraction performance. In this paper, we only use two features, entity types and parse trees generated by an English parser. Instead of designing new features, we explored methods to integrate these two features together by kernel composition and parse tree enrichment in clinical NLP.

## 2 Background

Relation extraction is the task of assigning one of many predefined relation types (e.g., Table. 1) to a relation in a given text. Relation extraction is commonly defined as detecting and classifying a relation between two named entities. The scope of a relation can be a sentence, a document or a text collection. The simplest task definition is extracting relations between two named entities appearing in a sentence together (Fig. 2-(b)) which is the definition used by the 2010 i2b2/VA Workshop on Natural Language Processing Challenges for Clinical Records [14]. The

2010 i2b2/VA challenge defines eight positive relation types (Table 1) for pairs of entities, of which the entity types are treatment, test, and medical problem.

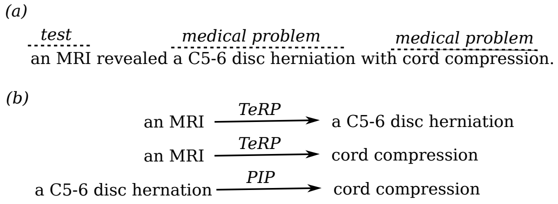

Figure 2: Relation extraction examples: (a) a target sentence with annotated concepts, (b) two “Test reveals medical problem (TeRP)” relations and a “Medical problem indicates medical problem (PIP)” relation are extracted.

As long as there is an annotated corpus, supervised machine learning is the preferred approach to address the relation extraction task. Frequently used methods can be categorized into two groups: feature-based and kernel-based. In open-domain NLP, feature-based methods [7, 6] dominated among earlier research which involved feature engineering as an important step. However, feature engineering is time-consuming and requires domain knowledge. To reduce the burden of feature engineering, kernel-based methods [16, 4, 2] were introduced to relation extraction. Zhang et al. [17] proposed a composite kernel-based approach that utilizes a convolution tree kernel [3] to capture complex syntactic information embedded in parse trees and an entity kernel to use lexical and semantic information. They reported their approach outperformed the state-of-the-art methods at the time in open-domain NLP.

The first attempt to apply supervised machine learning to clinical texts was done with 77 oncology narratives by Roberts et al. [13]. Their system utilized Support Vector Machine (SVM) classifiers with feature vectors made of lexical, syntactic and semantic features. The 2010 i2b2/VA challenge offered a corpus of 394 training reports, 477 test reports, and 877 unannotated reports. The top two performing systems [12, 5] from the challenge were also feature-based. In addition to features commonly used in open-domain NLP research, Rink et al. [12] used Wikipedia links and categories as a semantic relatedness indication between the two entities and edit distances between contextual strings to measure similarity between relations. de Bruijn et al. [5] used clinical domain semantic features utilizing domain knowledge sources such as the Unified Medical Language System (UMLS), manually built domain word/phrase dictionaries, and MEDLINE abstracts. They also used word cluster prefixes produced by the Brown clustering algorithm [1] and bootstrapped the training process with unannotated re-

| Positive relation type[14] | Training | Test |
|---|---|---|
| “Treatment improves medical problem (TrIP)” | 51 | 152 |
| “Treatment worsens medical problem (TrWP)” | 24 | 109 |
| “Treatment causes medical problem (TrCP)” | 184 | 342 |
| “Treatment is administered for medical problem (TrAP)” | 885 | 1732 |
| “Treatment is not administered because of medical problem (TrNAP)” | 62 | 112 |
| “Test reveals medical problem (TeRP)” | 993 | 2060 |
| “Test conducted to investigate medical problem (TeCP)” | 166 | 338 |
| “Medical problem indicates medical problem (PIP)” | 755 | 1448 |
| Total number of examples | 3120 | 6293 |

Table 1: Positive relation types of the 2010 i2b2/VA challenge and the number of training and test samples used

ports. The same research group that developed the latter system recently proposed a composite kernel-based system utilizing a convolution tree kernel and showed that it outperformed their previously top-ranked feature-based model [20].

In open-domain NLP, there has been research [17, 19, 18, 11] to integrate semantic features into parse trees so that a convolution tree kernel can explore the compound feature space of syntactic and semantic information. To the best of our knowledge, no prior work exists in using parse tree enrichment methods for clinical text. In this paper, we study the effects of various semantic features and different parse tree enrichment modes on accuracy, runtime, and model complexity on clinical notes from the 2010 i2b2/VA challenge.

# 3 Methods

## 3.1 Overview

In our experiment, an example (a relation) is represented by the entity types (test, treatment, or medical problem) of the two entities in a relation and a parse tree such as Fig. 3-(a). The final goal is to classify a relation using our trained SVMs into one of the eight positive relation types in Table 1. The most critical element in training SVMs and using them for classification in high dimensional space is to have an efficient and accurate similarity measure between two relations.

Our similarity measure is based on the similarity scores from two kernels: (1) the entity kernel and (2) the convolution tree kernel. The entity kernel compares the entity types of two relations to compute the similarity score. The convolution tree kernel searches common tree fragments (subtrees) (e.g., Fig. 3-(b)) between two parse trees to compute the similarity score. Then, the two similarity scores are linearly combined as the output of a composite kernel.

Our study focuses on improving the similarity measure computed by the convolution tree kernel by enhancing the parse tree structures with semantic information such as tree portion types and entity types. Three different tree portion types are used [18]: (1) a tree fragment belonging to the first entity, (2) a tree fragment belonging to the second entity, and (3) a tree fragment belonging to the connecting path between the first entity and the second entity. A parse tree is structurally modified in two different ways using semantic information, either inserting extra tree nodes carrying semantic information, or modifying the existing tree node labels with semantic information.

## 3.2 Composite kernel

We use one of the composite kernels proposed by Zhang et al. [17]. The composite kernel, $K_C$ is a linear combination of the normalized entity kernel $K_L^P$ and the normalized tree kernel $K_T$ (defined below): $K_C(R_1, R_2) = \alpha \hat{K}_L^P(R_1, R_2) + (1 - \alpha)\hat{K}_T(R_1, R_2)$ where $0 < \alpha < 1$, and $R_1$ and $R_2$ are two relations to be compared.

### 3.2.1 Entity kernel: $K_L$

$K_L(R_1, R_2)$ computes similarity between two relations ($R_1$ and $R_2$) using features of the two entities in each relation. Our study uses only the entity type feature with $K_L$. $K_L$ returns the sum of the number of common features between the first entities and between the second entities of $R_1$ and $R_2$. $K_L^P(R_1, R_2)$ is a polynomial expansion of $K_L$ of degree 2, which is $(K_L + 1)^2$. For example, the value from $K_L$ is higher when two relations share the same entity types for their first entities and the second entities than when two relations share the same entity type only for their first entities but not for the second entities.

### 3.2.2 Convolution tree kernel: $K_T$

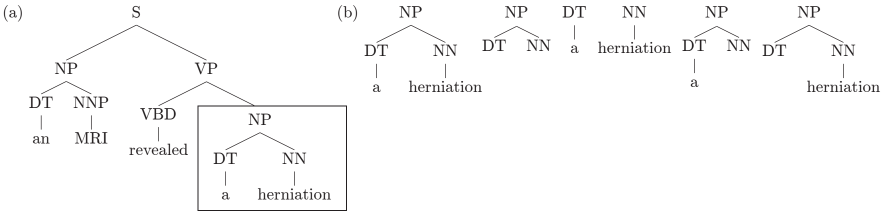

Figure 3: (a) An example parse tree (simplified from the previous example sentence used in Fig. 2). (b) All subtrees of the NP clause, "a herniation", that are compared for common subtree calculation.

We use the convolution tree kernel, $K_T$ described by Collins and Duffy [3]. $K_T(T_1, T_2)$ computes the similarity between two given parse trees, $T_1$ and $T_2$ from the two relations' texts. $K_T$ computes its score based on how similar the two trees are in terms of matches between their fragments (subtrees). Fig. 3-(b) shows what are the candidate subtrees to be considered for the boxed tree fragment in Fig. 3-(a). It is computationally expensive to enumerate all possible subtrees from two parse trees and compare those subtrees to find matches. To reduce such computational burden, in principle, $K_T$ counts the number of common subtrees under a pair of tree nodes, one each from $T_1$ and $T_2$ recursively. The recursive counting procedure adds $\lambda$ $(0 < \lambda < 1)$ to the similarity score if the productions (node labels) at two nodes being compared are the same and they are pre-terminals (nodes directly above the surface word) and adds 0 if the productions at two nodes are different. Otherwise, the procedure recursively repeats the same counting process with the child nodes where the return value is down-weighted by $\lambda$ as the recursive step goes deeper. Without down-weighting on each step in recursive counting along the trees, the convolution tree kernel can produce a very large value when it compares two large trees as opposed to two smaller trees even if the two smaller trees are more similar than the two larger trees. Thus, $\lambda$ is introduced to adjust the values produced by the convolution tree kernel to be comparable regardless of the tree size.

## 3.3 Tree enrichment

We compared six different enriched parse tree structures (two enrichment modes X three different semantic features). We chose the Shortest Path-enclosed Tree (SPT) as our baseline tree structure since it has performed the best in the experiments done in open-domain NLP [17] as well as in clinical domain NLP [20]. SPT is a shortest path subtree that encloses two entities in the parse tree. We then used the enriched trees as inputs to the convolution tree kernel in place of SPTs.

We consider two modes of parse tree enrichment: (1) inserting extra nodes (insertion) [17, 19, 9], and (2) modifying existing node labels (suffixing) [18]. We use three semantic features to label nodes in both enrichment modes: portion type, entity type, and the combination of portion type and entity type. Portion types are encoded by three letters: "A" for the first entity, "B" for the second , and "C" for the connecting path between the first entity and the second. Entity types are also encoded by three letters: "e" for "test", "r" for "treatment", and "p" for "medical problem". The combinations of portion types and entity types are encoded by combining two letters from the above encoding letters with a hyphen. For example, "A-e" means that the node belongs to the first entity of which type is "test".

Fig. 4-(c) is an example of the insertion mode. In this case, we use the combination of portion type and entity type to create labels for extra nodes we insert. "A-e" means that the tree fragment below it belongs to the first entity of which type is "test", and "B-p" means that the tree fragment below it belongs to the second entity of which type is "medical problem" Fig. 4-(b) and (d) are examples of the suffixing mode. The node labels of the tree depicted in (b) are suffixed with portion type, and the node labels of the tree in (d) are suffixed with entity type. Zhou and Zhu [18] proposed the suffixing mode for open-domain NLP and tested it with the portion type but not with entity type nor with the combined feature.

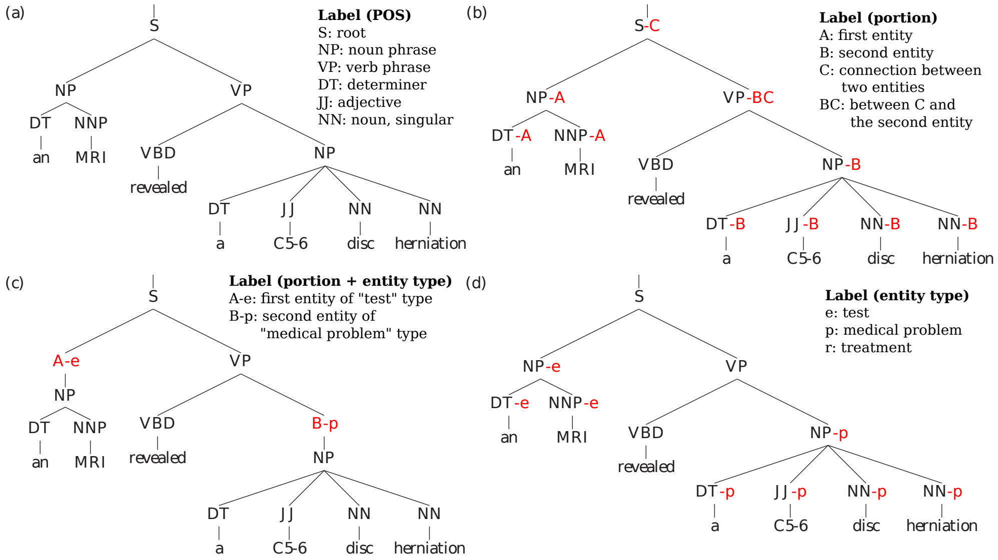

Figure 4: Parse tree enrichment examples: (a) Unmodified SPT. The non-terminal node labels are part of speech (POS) tags. (b) Portion type suffixes. POS tags are suffixed with portion types, and (c) Insertion of portion type + entity type nodes. Two nodes labelled with "A-e" and "B-p" are inserted. (d) Entity type suffixes. POS tags are suffixed with the entity type labels.

## 3.4 Training and classification

We train eleven binary classifiers in total for each of the eight positive relation types and three negative relation types. The positive relation types are relation types that the 2010 i2b2/VA challenge requires to be classified. The three negative relation types are "no relationship between a treatment and a medical problem" (NTrP), "no relationship between a test and a medical problem" (NTeP), and "no relationship between a medical problem and a medical problem" (NPP). Since the i2b2/VA challenge data do not provide training examples for these negative relation types, we generated negative training examples from the given entity records. We select all possible pair combinations of entities that appear in a sentence if one entity of the pair is a medical problem and the pair has not been identified as a positive relation instance in the challenge data. In summary, the training procedure for a relation type is as follows: (1)

generate negative examples, (2) obtain parse trees (SPTs) by running the Berkeley parser [10] on the positive and the negative examples, (3) produce enriched parse trees by introducing entity type and/or portion type to SPTs, and (4) train a SVM with examples of which features are entity types and enriched parse trees. We use the SVM-light-TK package [8] for binary SVM training and classification.

We used the one-versus-all strategy with eleven SVMs we trained to classify multiple class examples. Classification of a test example is done as follows: (1) get a parse tree using the Berkeley parser, (2) produce an enriched parse tree, (3) feed the example with entity types and the enriched parse tree to eleven SVMs, and (4) label the example with the relation type belonging to the SVM that produces the highest output value.

We use all relations that belong to a relation type as positive training examples and all other instances as negative training examples for a binary classifier.

# 4 Results

We evaluated our results with micro-averaged F1 (defined as $2 * P * R/(P + R)$), precision (P), and recall (R). Test examples from the three negative relation types (NTrP, NTeP, and NPP) were excluded from the computation of the evaluation metrics because the 2010 i2b2/VA challenge only requires the other eight positive relation types to be classified. We used the public "2010 Relations Challenge" dataset downloaded from the i2b2 web site. All six enriched parse tree structures improved the F1 score, P, and R compared to the baseline SPT structure. Among the enriched tree structures, the differences were marginal. Enrichment with entity type suffix showed the best performance in F1 score and recall. It was the fastest in runtime for training and test combined. Even though it resulted in 8.7% (1180) more support vectors in its models than SPT, it finished faster than SPT because of the reduced subtree search time.

| | SPT | Additional nodes | | | Suffixes | | |
|---|---|---|---|---|---|---|---|
| | | Portion type | Portion type + Entity type (Fig. 4-(c)) | Entity type | Portion type (Fig. 4-(b)) | Portion type + Entity type | Entity type (Fig. 4-(d)) |
| F1 | 0.761 | 0.7709 | 0.7701 | 0.7709 | 0.7713 | 0.7704 | 0.7725 |
| Precision | 0.8044 | 0.8126 | 0.8115 | 0.8109 | 0.814 | 0.8132 | 0.8131 |
| Recall | 0.7221 | 0.7332 | 0.7327 | 0.7346 | 0.7329 | 0.7319 | 0.7357 |
| # of support vectors | 13624 | 13608 | 12614 | 12334 | 15997 | 15781 | 14804 |
| Run-time (sec) | 474.99 | 482.37 | 468.39 | 478.42 | 477.42 | 484.78 | 467.9 |

Table 2: Performance of different parse tree structures

As shown in Table 1, TrAP, TeRP or PIP accounted for 83.3% of test examples. Thus, the accuracy with these three classes determined the overall accuracy of each enrichment scheme. We found that the entity type suffix tree produced the highest scores in precision (0.722), recall (0.4233) and F1 (0.5337) for PIP class, which explains the best overall F1 score among the seven tree structures. The F1 scores ranged from 0.8572 (SPT) to 0.8660 (entity type suffixes) for TrAP, from 0.9404 (SPT) to 0.9430 (portion type suffixes) for TeRP, and from 0.4891 (SPT) to 0.5337 (entity type suffixes) for PIP.

## 4.1 Semantic features

We tested three semantic features for tree enrichment – portion type, entity type, and the combination of both. When we compared three tree structures in each enrichment mode, the trees with entity type feature resulted in the least number of support vectors in the trained models in each enrichment mode. The entity type node tree has 9.3% fewer support vectors than the portion node tree, and the entity type suffix tree has 7.4% fewer support vectors than the portion suffix tree. Among the enriched parse tree structures in each enrichment mode, enrichment with entity type produced the highest F1 score with the least number of support vectors. The compound feature of portion type and entity type performed the worst among the three semantic features in each enrichment mode in both F1 score and recall. The number of support vectors is less than tree structures enriched with portion type and more than tree structures with entity type.

### 4.2 Enrichment modes

We found that among the two modes of enrichment, the suffixing mode can be potentially useful for reducing runtime. The portion suffix tree reduced runtime compared to the portion node tree by 1%, and the entity type suffix tree reduced the runtime by 2.2% compared to the portion node tree. This trend did not hold for the compound feature of portion type and entity type. The runtime reduction results were small because the current implementation for node label comparison uses a sequential character comparison resulting in longer runtime for longer labels. With better encoding of node labels to be equal length, we expect the runtime to be reduced further, especially in the classification time. Tree structures enriched by inserting nodes resulted in fewer number of support vectors compared to adding suffixes in all three semantic feature cases. For example, the entity type node tree has 20% fewer support vectors than the entity type suffix tree.

## 5 Discussion

Our evaluation with six different enriched parse tree structures showed that they improve the F1 score, precision and recall against the unmodified SPT. These results show that integrating semantic features into parse trees can improve relation extraction performance with tree kernel-based approaches in clinical domain NLP. For example, although the entity kernel already incorporate entity type information, we can still improve accuracy by integrating entity type information into the parse trees. We found it is feasible and desirable to utilize a compound feature space of semantic features and syntactic parse trees in relation extraction through convolution tree kernels.

Among three semantic features – portion type, entity types, and the combination of both – entity types produced the simplest trained model with the least number of support vectors which also can improve runtime performance. We observe that the differences in F1 scores between all enriched tree structures were relatively small. This is specific to the dataset, because for all of the relation types except PIP, portion type and entity type are interchangeable, because the 2010 i2b2/VA challenge specifies a relation should have a medical problem as the second entity. Except PIP relations, all relations have a treatment or a test as their first entity and a medical problem as their second entity. Therefore, just by knowing the entity type, the system can tell which subtree belongs to the first entity or to the second entity without any help from portion type. This can also explain why the compound feature of portion type and entity type produced a slightly worse F1 score than the other two semantic features. With the i2b2 dataset, adding portion type to entity-level features would have not added any additional information. Hence, for a better evaluation of compound semantic feature integration, combinations of independent or complementary semantic features would be desirable.

We showed that tree enrichment by suffixes can reduce runtime, and we believe that the runtime performance can be further improved by an efficient implementation for comparing node labels. For example, since we are dealing with a limited number of distinct node labels, we can map node labels to integer values and compare integer values instead of comparing every characters in two strings sequentially.

The main limitation of this work is that the dataset differs from the original challenge dataset which makes direct comparisons difficult. The original dataset was composed of clinical documents contributed by three different organizations: Partners Healthcare, Beth Israel Deaconess Medical Center, and the University of Pittsburgh Medical Center. However, the currently available data have fewer examples for training and classification because the data from the University of Pittsburgh Medical Center has been excluded. This is not a major flaw in the study itself since there are still data from two organizations exhibiting inter-organization variations. Since the tree convolution kernel compares structural similarity between parse trees that are generated by a parser governed by grammatical rules, the system is robust to inter-organization variations to some degree through linguistic abstraction. Although direct comparison is difficult, the dataset is still comparable. Our tree kernel-based system with tree enrichment achieved a higher F1 score than the two top performers [12, 5] of the 2010 i2b2/VA challenge and the recent Zhu et al's tree kernel-based system [20] which produced the F1 scores of 0.737, 0.731 and 0.742, respectively.

## 6 Conclusions

In this paper we have explored the feasibility of applying parse tree enrichment with semantic features to relation extraction from clinical notes. We found that all six different parse tree enrichment schemes performed better than non enriched parse tree (SPT) in F1, precision, and recall. While we only used two features, entity types and a parse tree, the performance was comparable to the state-of-the-art results. Since entity types are usually pre-annotated and syntax parsing is necessary for syntactic features extraction, our approach has an advantage over models with more features in terms of the time to extract features. With fewer features, it also has fewer sources of error propagation that can be introduced in the feature extraction step.

## Acknowledgments

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ICT Consilience Creative Program (IITP-2015-R0346-15-1007) supervised by the IITP(Institute for Information & communications Technology Promotion).

## References

[1] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-Based n-gram Models of Natural Language. *Computational linguistics*, 18(4):467–479, 1992.

[2] Razvan C Bunescu and Raymond J Mooney. Subsequence Kernels for Relation Extraction. In *Advances in neural information processing systems*, pages 171–178, 2005.

[3] Michael Collins and Nigel Duffy. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems 14*, pages 625–632. MIT Press, 2001.

[4] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proc. of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*, number Table 2, pages 423–es, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[5] Berry de Bruijn, Colin Cherry, Svetlana Kiritchenko, Joel Martin, and Xiaodan Zhu. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *JAMIA*, 18(5):557–562, 2011.

[6] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics ACL 05*, (June):427–434, 2005.

[7] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proc. of the ACL 2004 on Interactive poster and demonstration sessions*, page 22, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[8] Alessandro Moschitti. A study on convolution kernels for shallow semantic parsing. In *ACL '04: Proc. of the 42nd Annual Meeting on Association for Computational Linguistics*, page 335, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[9] Tvt Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proc. of the 2009 Conf. on Empirical Methods in NLP*, number August in EMNLP '09, pages 1378–1387, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[10] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proc. of the 21st Intl. Conf. on Computational Linguistics and the 44th annual meeting of the ACL - ACL '06*, number July, pages 433–440, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[11] Barbara Plank and Alessandro Moschitti. Embedding Semantic Similarity in Tree Kernels for Domain Adaptation of Relation Extraction. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1498–1507. ACL, 2013.

[12] Bryan Rink, Sanda Harabagiu, and Kirk Roberts. Automatic extraction of relations between medical concepts in clinical texts. *JAMIA*, 18(5):594–600, 2011.

[13] Angus Roberts, Robert Gaizauskas, Mark Hepple, and Yikun Guo. Mining clinical relationships from patient narratives. *BMC bioinformatics*, 9 Suppl 11(June):S3, 2008.

[14] Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *JAMIA*, 18(5):552–6, 2011.

[15] Stephen T Wu, Hongfang Liu, Dingcheng Li, Cui Tao, Mark a Musen, Christopher G Chute, and Nigam H Shah. Unified Medical Language System term occurrences in clinical notes: a large-scale corpus analysis. *JAMIA*, 19(e1):e149–156, June 2012.

[16] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel Methods for Relation Extraction. *Journal of Machine Learning Research*, 3:1083–1106, 2003.

[17] Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. A Composite Kernel to Extract Relations Between Entities with Both Flat and Structured Features. In *Proc. of the 21st Intl. Conf. on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 825–832, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[18] Guo-Dong Zhou. Kernel-Based Semantic Relation Detection and Classification via Enriched Parse Tree Structure. *Journal of Computer Science and Technology*, 26(1):45–56, 2011.

[19] Guodong Zhou, Longhua Qian, and Jianxi Fan. Tree kernel-based semantic relation extraction with rich syntactic and semantic information. *Information Sciences*, 180(8):1313–1325, April 2010.

[20] Xiaodan Zhu, Colin Cherry, Svetlana Kiritchenko, Joel Martin, and Berry de Bruijn. Detecting concept relations in clinical text: Insights from a state-of-the-art model. *Journal of Biomedical Informatics*, 46(2):275–285, April 2013.