A survey of kernel methods for relation extraction

Guillermo Moncecchi[1,2], Jean-Luc Minel[2], and Dina Wonsever[1]

[1] Instituto de Computación, Facultad de Ingeniería
Universidad de la República
Montevideo
Uruguay
[2] Laboratoire MoDyCo, UMR 7114 CNRS
Université Paris Ouest
Nanterre La Dfense
France

Abstract. In this paper we present the main kernel approaches to the problem of relation extraction from unstructured texts. After a brief introduction to the problem and its characterization as a classification task, we present a survey of the methods and techniques used, and the results obtained. We finally suggest some future lines of work, such as the use of information retrieval techniques and the development of event factuality identification methods.

1 Introduction

Relation extraction is a task within the field of *information extraction*. It involves the identification of relations between entities already identified in natural language texts. Two subtasks can be considered: we may just want to discover if two or more candidate entities are related (the subtask of *relation detection*), or we may wish to know which of a predefined set of relations hold between them (the subtask of *relation characterisation*).

In the sentence *"This cross regulation between Drosha and DGCR8 may contribute to the homeostatic control of miRNA biogenesis"*, where the proteins Drosha and DGCR8 are mentioned, we can identify a CROSSREGULATION relation between them. In the molecular biology domain, the investigation of protein-protein interaction networks plays a key role in the construction of knowledge about molecular mechanisms. Because stating hand-curated relations in the appropriate databases is a very time-consuming task, the application of relation extraction techniques to the rapidly growing amount of information that is available in the research literature can undoubtedly help domain researchers.

Relation extraction can be characterised as a classification problem: if we consider pairs (or even n-uples) of entities that could be related, we just need to determine if they are indeed related (which is a problem of binary classification) or even to determine *which* relation holds between them (which is a problem of n-class classification).

In this paper, we present a survey of kernel approaches to relation extraction. In the MUC-7 Conferences, where the relation detection and characterization tasks were first formulated, all but one of the systems (Miller et al., 1998) were based on handcrafted rules. In machine learning approaches, patterns for identifying relations are not manually written but are learned from labelled examples. While it may be difficult to generate enough labelled examples (a manual and time-costly task), machine learning solutions have shown in many different tasks their ability to adapt to different domains and solve problems that handcrafted rules could not.

In the following sections, we present the main techniques and methods, from feature-based to state-of-the-art tree and graph kernel methods and their combination, showing the results of their application to comparable evaluation corpora. We suggest some future lines of work, including the incorporation of information retrieval techniques to the task and the development of event factuality identification methods.

2 Kernel methods

Most machine learning algorithms are feature-based. Feature-based methods represent labelled examples as a sequence $f_1, f_2, \ldots, f_m$ of features, living in an m-dimensional space. For example, in the relation extraction task we can consider a sentence as an example, represented by a list of binary attributes, one for each possible token, indicating if the sentence includes that particular token or not.

The problem with feature-based methods is that sometimes data cannot be easily represented with explicit feature vectors (for example, natural language sentences are better described by means of trees or even graphs). In those cases, feature extraction is a very complex task, and leads to very high dimensional vectors, which in turn leads to computational problems. Kernel-based methods try to solve this problem by *implicitly* calculating feature vector dot-products in very high dimensionality spaces, without having to make each vector explicit.

In kernel methods, labelled examples are not necessarily feature vectors. Instead, a similarity function (or kernel) between examples is computed and discriminative methods are used to label new examples. A kernel function over an object space X is a symmetric, positive semi-definite binary function $K : X \times X \rightarrow [0, \infty]$ that assigns a similarity score between two instances of X. An important property of kernels is that if we have a collection $f_1, f_2, \ldots, f_n$ of features, where $f_i : X \rightarrow R$, the dot product between two vectors is necessarily a kernel function; the converse also holds.

There are many learning algorithms, from the simple Perceptron algorithm (Rosenblatt, 1958) to Voted Perceptron (Freund and Schapire, 1999) and Support Vector Machines (Cortes and Vapnik, 1995) that can be represented in what is called the *dual form*, which just relies on dot products between examples. In those cases, dot products can be replaced by kernel functions (the "kernel trick"). This allows us to compute, through kernels, the dot product of certain feature vectors, without necessarily enumerating all the features (for example, (Lodhi et

al., 2000) defined a kernel to compute in polytime the number of common subsequences in two strings, a problem with an exponential number of features). This allows for the implicit exploration of a much larger feature space than feature-based learning algorithms. For a detailed explanation on how kernel methods work, see (Cristianni and Shawe-Taylor, 2000).

Kernel methods for relation extraction were first introduced by (Zelenko et al., 2003). They proposed this kind of machine learning methods after their successful application to similar problems, such as natural language parsing (Collins and Duffy, 2001). In this section we survey their problem formalization and kernels, then discuss other approaches that improved classification performance over comparable corpora.

Most of the methods here presented were evaluated on the ACE corpus, a 300K news corpus annotated with entities and relations, created by the Linguistic Data Consortium for the Automatic Content Extraction Program (Doddington et al., 2004), and the AImed corpus (Bunescu and Mooney, 2005), a molecular biology corpus consisting of 225 Medline abstracts, annotated with human proteins and their interactions.

2.1 First kernel approaches to relation extraction

(Zelenko et al., 2003) reduced the problem of relation extraction to the problem of pair-of-entities classification: examples consisted of parts of sentence shallow parse trees, where relation roles (for example: member, or affiliation) were identified and expressed by tree attributes. For training, examples were marked with {+1,-1} labels, expressing wheter the tree linked roles in the examples were indeed semantically related. Figure 1 shows one of the positive examples built from the shallow parse tree of the sentence *"John Smith is the chief scientist of the Hardcom Corporation"*

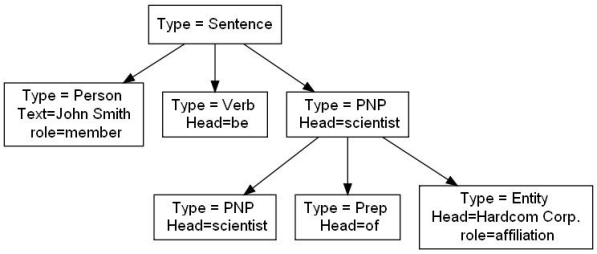

Fig. 1. Example of the person-affiliation relation (from (Zelenko et al., 2003))

They defined a similarity function between example trees that took into account the number of similar subsequences of children with matching parents. They showed that this similarity function was a kernel, and could therefore be

used in any dual-representable learning algorithm. They worked with two different types of kernels: contiguous sub tree kernels (where the similarity measure enumerated only contiguous subsequences of children), and the general case of sparse sub tree kernels. In both cases they gave a fast algorithm for computing the similarity function ($O(mn)$ for the case of contiguous sub tree kernels, and $O(mn^3)$ for the case of sparse sub-tree kernels, where m and n are the number of children in the two examples).

As working in such a large feature space could easily lead to overfitting, they evaluated their approach using two different kernel algorithms: Support Vector Machines and Voted Perceptron, implementing both kernels in each case. They compared them with three feature-based algorithms: Naive-Bayes, Winnow and SVM (where features were conjunctions of conditions over relation example nodes). They found that kernel methods outperformed feature-based methods in almost every scenario, achieving a best F-measure of 86.8 for the person-affiliation relation, and 83.3 for the organization-location relation, in both cases using Support Vector Machines with a sparse tree kernel.

2.2 Feature-based kernel approaches

In feature-based kernels, the dot-product between feature vectors is explicitly calculated. These methods use a similar approach to traditional feature-based machine learning methods, but they can also exploit some interesting properties of kernels (for instance, a product or sum of kernels is also a kernel).

(Zhao and Grishman, 2005) used feature-based kernels, with information from tokenization, parsing and deep dependency analysis, extending some of them to generate high order features, and composing them in different ways. (Zhou et al., 2005), using SVM, incorporated diverse lexical, syntactic and semantic knowledge features. They argued that full parsing information was not very important, because most of the relations were within a short distance in their corpus. They also showed that the use of WordNet and Name Lists could improve classification results. Based on their work, (Wang et al., 2006) added POS tags and several general semantic features, using a semantic analyser and WordNet. They found that basic features (those arising from words, POS tags, entity, mention and overlap) were far more important than the deeper ones (chunk, dependency tree, parse tree and semantic analysis).

(Bunescu and Mooney, 2005) observed that the information required to assert a relation between two entities could be captured by the shortest path between the two entities in the dependency graph. Based on this, they developed a simple kernel which incorporated words and word class features of the path components, and calculated the number of common features between two relation examples. (Erkan et al., 2007) adapted their work to the domain of protein-protein relation extraction, measuring the similarity of two examples by comparing their corresponding shortest path using cosine similarity and edit distance. Using semi-supervised models on top of dependency features and using harmonic functions (a semi-supervised version of the kNN classification method) and transductive

SVMs, they showed that semi-supervised approaches could improve classification performance.

2.3 Tree, graph and combined kernel methods

Instead of directly computing the dot product between examples, and working on the same hypothesis as (Zelenko et al., 2003) (i.e. that instances containing similar relations shared similar syntactic structure), other kinds of kernels have been developed: they work with instances represented by trees or even graphs, instead of just feature vectors.

(Culotta and Sorensen, 2004) used dependency trees as representations of relation examples. They augmented these trees with features in each node (including word, part of speech, entity type and Wordnet hypernyms), trying to incorporate more information for classification, and used a slightly more general version of (Zelenko et al., 2003) kernels.

(Bunescu and Mooney, 2006), used a subsequence kernel that computed the number of common subsequences of words and word classes between examples (considering only those subsequences where candidate entities existed, and words belonged to three predefined patterns). (Giuliano et al. 2006) used the same patterns to discover the presence of a relation, but each pattern was represented as a bag-of-words, instead of sparse subsequences, adding n-grams to improve classification performance. Another kernel, the *Local Context Kernel* added information about the local contexts of the candidate interacting entities.

(Zhang et al., 2006) combined a feature-based entity kernel (which measured the number of common features between two entities), and a convolution parse tree kernel (which counted the number of common sub-trees between two relations), in two different ways: as a linear combination, and as a polynomial expansion that aimed to explore the combined features from the first and second entities of the relationship. They also systematically explored which parts of the parse tree could be used for similarity calculation. They obtained their best results using the sub-tree enclosed by the shortest path linking two involved entities in the parse tree, combined via polynomial expansion with the entity kernel.

(Zhou et al., 2007) tried to improve on the Collins and Duffy convolution kernel, proposing what they called a *context-sensitive convolution tree kernel*. This method first automatically determined a dynamic context-sensitive tree span (the original convolution kernels were context free: a sub tree did not consider context information outside the sub tree), and then used not only the found tree, but also its ancestor node paths as contexts for calculating the similarity. Similar to the previous work, they combined their kernel via polynomial interpolation with the linear kernel described in (Zhou et al. , 2005), achieving a state-of-the-art F-measure of 74.1 using a composite kernel.

(Airola et al., 2008), in their work on protein-protein interaction proposed a graph kerne on dependency parses. They defined a weighted, directed graph, composed of two unconnected subgraphs: one with the dependency structure of the sentence, and the other one with just the linear order of the words (using

word, POS and entity information, entity information and indicator of their relative position with respect to candidate entities). On this graph, they defined what they called the *all-dependency-paths* kernel, that computed the summed weights of all possible paths connecting two vertices.

For the sake of comparison, table 1 presents precision, recall and F-measure for some of the presented methods on the Automatic Content Extraction 2003 (numbers without parenthesis) and 2004 corpora, for the tasks of relation detection, relation characterization for the top high-level relation types. Table 2 presents results on the AImed corpus.

Table 1. Relation classification performance on the ACE corpus

	Relation Identification			Relation Types Characterization		
	P	R	F	P	R	F
(Culotta and Sorensen, 2004)	81.2	51.8	63.2	67.1	35.0	45.8
(Zhao and Grishman,2005				(69.2)	**(70.5)**	**(70.3)**
(Bunescu and Mooney, 2005)				65.5	43.8	52.5
(Zhou et al., 2005)	**84.8**	**66.7**	**74.7**	77.2	60.7	68.0
(Bunescu and Mooney, 2006)				73.9	35.2	47.7
(Wang et al.,2006)	**(73.9)**	**(69.5)**	**(71.6)**	**(71.4)**	(60.0)	(65.2)
(Zhou et al.,2007)				**80.8**	**68.4**	**74.1**

Table 2. Relation characterization performance on the Aimed corpus

	Precision	Recall	F-measure
(Giuliano et al.,2006)	60.9	57.2	**59.0**
(Bunescu and Mooney,2006)	**65.0**	46.4	54.2
(Airola et al.,2008)	52.9	**61.8**	56.4

3 Conclusions

As this paper shown, extensive work has been done on the task of relation extraction. Kernel-based methods present many features that makes them specially suitable for this kind of tasks: they can accomodate features from different analyses (lexical and syntactic analysis, information from external sources); the supervised learning classifiers they use (Support Vector Machines, Voted Percpetron) are known for their good performance even when few training data is available; finally, their ability to represent similarity measures between complex

structures allows them to incorporate information not easily represented using the traiditional feature-value pairs (such as dependency or shallow parses).

From the results, it is not clear which type of kernels (those computed as an explicit dot-product between feature vectors or directly calculated from the original structures, being them strings, trees or graphs) are better for the relation-extraction task, nor which kernel could yield better performance using the same algorithm. However, later work seems to indicate that combining many sources of information and different kernels can indeed improve performance, accommodating smoothly a large amount of linguistic features, including entity related semantic information, syntactic parse and dependency trees and lexical information.

The huge amount of unannotated texts for some domains suggests that the incorporation of semi-supervised approaches and the adaptation of information retrieval techniques could lead to precision and recall improvement. For example, having hypothesized that a relation holds between two proteins, we could gather more information, based on their sentence and document co-occurrence on unannotated texts to improve the precision of the hypothesis.

In the highly specialized domain of molecular biology, there has recently been considerable research effort towards ontology-based annotation of entities and relations on natural language texts. The availability of annotated corpora plays a key role in the success of any supervised machine learning task. Every possible model is the result of inference reasoning of some sort of previously seen annotated data. Two annotated corpus (not mentioned in this survey) specifically oriented to the relation extraction task in the biomedical domain, have been published: the Bioinfer corpus (Pyysalo et al., 2007) and the Genia Event corpus (Kim et al. 2008).

Relation extraction would also benefit from advances in general semantics and pragmatic recognition tasks in natural language processing. Contextual features such as polarity or modality clearly may change local inferences about the factuality status of an event or extracted relation, and they should be considered (Sauri et al., 2006) .We think that the combination of kernel-based methods with semantic features produced by careful studies of event modality applied to the molecular biology domain (involving the work of biologists, linguists and natural language processing specialists) could lead to successful results.

All these tasks could undoubtedly contribute to the automatic extraction and even inference of previously unseen relations, which could be the basis for subsequent experimental methods of validation.

References

Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., Salakoski, T.: A graph kernel for protein-protein interaction extraction. In: BioNLP (2008), http://www.aclweb.org/anthology-new/W/W08/W08-0601.pdf

Bunescu, R., Mooney, R.: Subsequence kernels for relation extraction. In: Weiss, Y., Schölkopf, B., Platt, J. (eds.) Advances in Neural Information

Processing Systems 18, pp. 171–178. MIT Press, Cambridge, MA (2006), http://books.nips.cc/papers/files/nips18/NIPS2005_0450.pdf

Bunescu, R.C., Mooney, R.J.: A shortest path dependency kernel for relation extraction. In: Proceedings of the Joint Conference on Human Language Technology / Empirical Methods in Natural Language Processing (HLT/EMNLP). pp. 724–731. Association for Computational Linguistics, Vancouver, British Columbia, Canada (October 2005), http://www.aclweb.org/anthology/H/H05/H05-1091.pdf

Collins, M., Duffy, N.: Convolution kernels for natural language. In: Advances in Neural Information Processing Systems 14. pp. 625–632. MIT Press (2001)

Cortes, C., Vapnik, V.: Support vector networks. In: Machine Learning. pp. 273–297 (1995)

Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press (March 2000), http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0521780195

Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (2004), http://acl.ldc.upenn.edu/P/P04/P04-1054.pdf

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., Weischedel, R.: The Automatic Content Extraction (ACE) Program–Tasks, Data, and Evaluation. Proceedings of LREC 2004 pp. 837–840 (2004)

Erkan, G., Ozgur, A., Radev, D.R.: Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). pp. 228–237 (2007), http://www.aclweb.org/anthology/D/D07/D07-1024

Freund, Y., Schapire, R.E.: Large margin classification using the perceptron algorithm. In: Machine Learning. pp. 277–296 (1999)

Giuliano, C., Lavelli, A., Romano, L.: Exploiting shallow linguistic information for relation extraction from biomedical literature. In: 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL '06). pp. 401–408. European Chapter of the Association for Computational Linguistics, Trento, Italy (April 2006), http://acl.ldc.upenn.edu/E/E06/E06-1051.pdf

Guodong, Z., Jian, S., Jie, Z., Min, Z.: Exploring various knowledge in relation extraction. In: ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 427–434. Association for Computational Linguistics, Morristown, NJ, USA (2005), http://dx.doi.org/10.3115/1219840.1219893

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

Lodhi, H., Taylor, J.S., Cristianini, N., Watkins, C.J.C.H.: Text classification using string kernels. In: NIPS. pp. 563–569 (2000), http://citeseer.ist.psu.edu/lodhi02text.html

Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schwartz, R., Stone, R., Weischedel, R., Group, T.A.: Algorithms that learn to extract information BBN: Description of the Sift system as used for MUC-7. In: MUC-7 (1998), http://acl.ldc.upenn.edu/muc7/M98-0009.pdf

Pyysalo, S., Ginter, F., Heimonen, J., Bjorne, J., Boberg, J., Jarvinen, J., Salakoski, T.: Bioinfer: A corpus for information extraction in the biomedical domain. BMC Bioinformatics 8(1) (2007), http://dx.doi.org/10.1186/1471-2105-8-50

Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain pp. 89–114 (1958), http://portal.acm.org/citation.cfm?id=65669.104386

Wang, T., Li, Y., Bontcheva, K., Cunningham, H., Wang, J.: Automatic extraction of hierarchical relations from text. pp. 215–229 (2006), http://dx.doi.org/10.1007/11762256_18

Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. J. Mach. Learn. Res. 3, 1083–1106 (2003), http://portal.acm.org/citation.cfm?id=944919.944964

Zhang, M., Zhang, J., Su, J., Zhou, G.: A composite kernel to extract relations between entities with both flat and structured features. In: ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL. pp. 825–832. Association for Computational Linguistics, Morristown, NJ, USA (2006), http://dx.doi.org/10.3115/1220175.1220279

Zhao, S., Grishman, R.: Extracting relations with integrated information using kernel methods. http://acl.ldc.upenn.edu/P/P05/P05-1052.pdf

Zhou, G., Zhang, M., Ji, D., Zhu, Q.: Tree kernel-based relation extraction with context-sensitive structured parse tree information. In: Proceedings of EMNLP-CoNLL 2007 (2007), http://acl.ldc.upenn.edu/D/D07/D07-1076.pdf