Full Syntactic Parsing for Enrichment of RDF dataset

Michel Gagnon, Caroline Barrière and Eric Charton
michel.gagnon@polymtl.ca, caroline.barriere@crim.ca, eric.charton@polymtl.ca

École Polytechnique de Montréal, Centre de Recherche Informatique de Montréal,
http://www.polymtl.ca/gigl http://www.crim.ca

Abstract. RDF data extracted automatically often contain long textual literals. This paper shows how to use natural language processing techniques to automatically generate specific RDF triples from the information in the literals. We look specifically at drug indications found in the DailyMed dataset. We develop knowledge schemas to capture its information as well as precise syntactic-based methods of knowledge extraction to automatically generate instances of these schemas from textual data.

1 Introduction

From a Natural Language Processing point of view, we can approach the Semantic Web as if it were a large resource of semi-structured textual data on which knowledge extraction techniques can be applied. Many datasets (e.g. Drugbank, DailyMed, Diseasome, Medicare, SIDER[1]) were created by a RDFization process of databases or semi-structured documents. Their generation, achieved by a set of ad hoc rules, results in a dataset that contains precise data (obtained from databases), and long literals (obtained from texts).

Natural language processing techniques can be used to automatically extract information from the literals. We show how a limited manual effort invested in defining knowledge schemas and text-based knowledge extraction rules can be of value to the Semantic Web community, by the fact that it could help in the RDFization process of many datasets.

We look more specifically at the DailyMed dataset. The site DailyMed[2] published by the National Library of Medicine provides high quality information about market drugs. The RDF version of DailyMed is part of the Linking Open Drug Data project [1]. It describes about 3600 drugs and provides many predicates. Some predicates in the RDF view link to resources, and others to literals of variable sizes. Predicates such as *adverseReaction*, *clinicalPharmacology*, *precaution* or *indication* lead to literals on which text analysis techniques can be used to further pursue the RDFization. In this research, we focus on the indication

[1] http://www4.wiwiss.fu-berlin.de/ provides access to Drugbank, DailyMed, Diseasome, Medicare and SIDER.
[2] http://dailymed.nlm.nih.gov

predicate. The indication for each drug is rather lengthy with its size varying from 1 word to 1338 words (average of 127 words).

The texts found in the labels contain sentences like *Fluticasone propionate ointment is a medium potency corticosteroid indicated for the relief of the inflammatory and pruritic manifestations of corticosteroid-responsive dermatoses in adult patients.* The word *indicated* appears to be a strong linguistic pattern for expressing a drug's indication. As we will present in more details in section 2 (related work), linguistic patterns are often used for knowledge extraction. But, in our case, for an in-depth analysis of these indication predicates, they are not sufficient. Considering the complexity of language and the complexity of the knowledge to be represented, we chose to explore deep syntactical language analysis approaches and to define a knowledge schema for knowledge representation for drug therapy.

This paper is structured as follow. After a brief overview of related work, we define a general method for knowledge extraction and a general framework for knowledge representation as applied to the DailyMed indication predicate. Then we look more deeply at the resource and refine both our knowledge schema and our knowledge extraction method. Finally, we present an evaluation of our extraction approach.

2 Related work

Numerous proposals have been made to facilitate ontological engineering through automatic discovery from domain data or domain-specific natural language texts. Early algorithms for relation extraction, like DIPRE [3], SNOWBALL [4], only rely on simple string-based regular expressions to recognize relations such as author-book or expression patterns over words and named entity tags. Such pattern-based techniques of information extraction are still in use for some semantic web applications when the text content is structured enough to allow extraction of data intended to populate an ontology. DBPedia, for example [5], uses pattern matching techniques to recognize the structure of Infoboxes in Wikipedia and collect data. The algorithm used can detect lists of objects, which are transformed to RDF lists structured in an ontology.

At least three dominating machine-learning related paradigms have been applied to the task of extracting relational facts from non-labelled or structured text [6]. *Supervised approaches*, where sentences in a corpus are first hand-labelled for the presence of targeted pair of entities and facts and the relations between them. A machine learning technique (e.g. Support Vector Machine [7] or Markov Logic Network reasoner [8]) is then used to learn the relation and generate a model to discover new relations in non-annotated texts. *Unsupervised information extraction* approaches, alternatively, extract strings of words between entities in large amounts of text, and then cluster and simplify these word strings to produce relation-strings. Unsupervised approaches use very large amounts of data and extract very large numbers of relations, but the resulting relations include generally an important amount of non relevant discovery, that

make the results difficult to map to a particular knowledge base[9]. *Bootstraping* techniques are used with a very small number of seed instances or patterns to do boot-strap learning with a large corpus and to extract a new set of patterns. Those patterns are used to extract more instances, which are used to extract more patterns, iteratively. However, the resulting patterns mostly contains a lot of noisy information and suffer from low precision[10].

Recently, [6] proposed the *Distant Supervision* algorithm, supervised by a database rather than by labeled text. They use Freebase[3] as a large semantic database, to provide supervision information for relation extraction, and investigate the value of syntactic features in their system. Those modern Information Extraction Systems are evaluated trough standard evaluation campaign like NIST Knowledge Base Population (KBP) from Text Analysis Conference (TAC) [11], ACE [12] or BioNLP [13][14].

Some specific Information Extraction techniques are investigated for the Biomedical field[15]. This field is certainly the one in which more precise syntactic-based approaches are used and promoted[13][14]. For example, an approach aimed at automating the process of extracting functional relations e.g. interactions between genes and proteins) from biomedical literature using syntactic features have been studied in [16]. We would situate our present work within that same paradigm of investing manual effort to help the precise automatic discovery of relations, either gene interaction, or drug-disease interaction, or other.

3 General approach

Our first task is to define a knowledge schema for the representation of the knowledge expressed in the indication predicate. We start with a general schema which we will refine after (see section 4). Associated with such schema, we define a general extraction method to automatically extract information from the indication object (the literal).

3.1 Knowledge schema

We first devise a general Treatment logical structure, that can be expressed in RDF, and that includes the concepts of drug, disease, and treatment as well as some refinements on the type of treatment. Below is an indication sentence and the knowledge structure we wish to automatically extract.

Restoril is indicated for the short-term treatment of insomnia.

```
dailymed_drug:3239 rdfs:label "Restoril".
loc:T234 loc:drugInvolved dailymed_drug:3239;
         rdf:type loc:Treatment, loc:DrugTherapy, loc:ShortTermTreatment;
         loc:target "insomnia";
```

Within the semantic web paradigm, we use the RDF format, and we express concepts via resources (URIs) and relations between concepts via predicates.

[3] www.freebase.com

Our vocabulary for the RDF descriptions consists in a set of classes that are used to specify treatment types. To clearly indicate that the treatments involve some drug, we attribute to all of them the type `loc:DrugTherapy`. We use the predicate `loc:target` to link the treatment with its target disease or disorder. Predicate `loc:drugInvolved` identifies the drug that is used for the treatment[4].

We do not make attempts in this research to link the extracted information to existing URIs. This is a research problem in itself outside the scope of the present research.

3.2 Knowledge extraction

For precise knowledge extraction, we use a syntactic approach. As mentioned earlier in section 2, this is in line with recent work on knowledge extraction in the biomedical domain. The text is parsed with Stanford Parser[5], a statistical parser that provides a dependency tree output. In a dependency tree, every word, except the root word (which usually is the main verb of the sentence) is linked to another word by a dependency relation. In a dependency relation, one word (the head word) is dominating the other one. Figure 1 shows the dependency tree returned by Stanford parser for our example sentence *Restoril is indicated for the short-term treatment of insomnia* (note that the root node is indicated in boldface). There is a subject relation (nsubjpass) between *Restoril* and *indicated*, and an indirect object relation (prep) between *indicated* and *for*. Note that the actual object is *treatment*, the dependent of *for*.

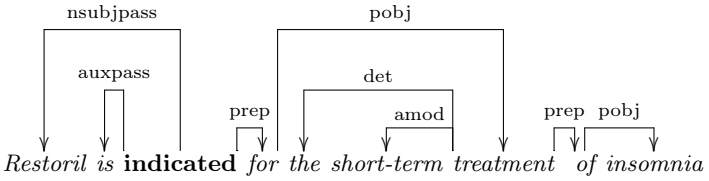


Fig. 1. Example of dependency tree.

Let T be a dependency tree for sentence S and $sub(Head)$ the sub-tree of T whose root node is the word $Head$ in S.

The extractor is based on a set of rules $\langle Pattern, Extraction \rangle$, where $Pattern$ is a syntactic pattern, in fact a dependency tree where some nodes are variables to be substituted by a sub-tree, and $Extraction$ is a specification of how the RDF

[4] To be complete, the drug therapy description should take into account dosage forms and route of administration. In this paper, we leave these aspects out, but the method that is presented could be extended to embrace them.

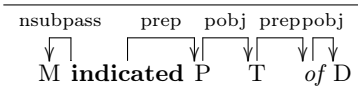
[5] http://nlp.stanford.edu/software/lex-parser.shtml

triples are to be generated from the information found in the dependency tree that will be matched with the pattern. We define functions $Drug(U_1, U_2, subTree)$, $Disease(URI, subTree)$ and $Treatment(URI, subTree)$, that are used in the extraction part of a rule, to extract information for drug, disease and treatment, respectively. The function for *Drug* for the example above would be:

$Drug$(`dailymed_drug:3239`, `loc:T234`, sub(Restoril))
= `loc:T234 loc:drugInvolved dailymed_drug:3239 .`
`dailymed_drug:3239 rdfs:label "Restoril" .`

An additional function, $Qualif(URI, sub)$, is used to process the following qualifiers: *short-term*, *long-term*, *first-line*, *second-line*, *initial* and *acute*. These qualifiers are not processed in the $Treatment()$ function because they do not strictly subcategorize a treatment, as opposed to a *palliative treatment* which is a specific type of treatment. One important consequence of this separation is that a qualifier may be combined with any type of treatment. We will see subcategorization of treatments in section 4 as we refine our schema.

Pattern



$P \in$ { in, for, as }

$T \in$ {treatment, therapy, prevention, control, relief, management, maintenance, prophylaxis}

Extraction

$Drug(U_D, U_T, subtree(M)) \cup Treatment(U_T, sub(T)) \cup Disease(U_T, sub(D)) \cup Qualif(U_T, sub(T))$

Fig. 2. General extraction rule.

Figure 2 shows the most generic (default) extraction rule used in our system. It calls for each sub-tree the appropriate extraction function that will generate a partial RDF description. These partial descriptions are then merged to get the complete RDF description. Note that URIs corresponding to drug and treatment, that is, U_D and U_T respectively, are given as parameters to the functions.

4 Refinement inspired by corpus analysis

In this section, we see how a simple frequency analysis on the corpus made of all drug indications show lexical and syntactical variations that help us refine our representation schema and consequently our extraction rules.

4.1 Corpus statistics

All literals, object of the indication predicate in DailyMed have been joined to form a corpus of 3683 indications. Frequencies of 5-grams were counted to find variation in expression. Table 1 shows the most frequent 5-gram. We see clearly that the relation of interest is almost always expressed by the passive form *is indicated for/in/as*. Note that there are many occurrences of negation, where it is specified that some drug is not indicated for some disease. It is important to recognize these negated forms, otherwise we would extract an indication description where we should not.

5-gram	# occ.
indicated for the treatment of	1311
indicated in the treatment of	354
indicated for the relief of	346
reduce the development of drug-resistant	284
to treat or prevent infections	278
alone or in combination with	173
the treatment of patients with	164
indicated for the management of	163
indicated as an adjunct to	142
indicated for the prevention of	90
to reduce the risk of	76
for the topical treatment of	72
is not indicated for the	69
indicated as adjunctive therapy in	68
adjunctive therapy in the treatment	55

Table 1. Excerpt of the extracted 5-grams.

It is also clear that the interaction between a drug and a disease is not limited to a generic treatment relation. Reality is more complex: treatments can be subcategorized (prophylactic, palliative, short-term, symptomatic, management, etc), some treatments involve drug combination and, finally, some are adjunctive therapies, that is, they are given in addition to an initial treatment. From our corpus analysis, we identified more than 30 kinds of treatment. Examples of these are listed in Table 2. In each row, we indicate the category of the treatment, a short definition and the number of occurrences found in the corpus. Although unspecified treatments are most commonly found in Dailymed descriptions, more than half of indications lead to more specific treatments. That number is not negligible and confirms the relevance of recognizing these specificities.

4.2 Refinement of the representation schema

Based on Table 2, different variations need to be included in the representation schema. We show this refinement process with one example: adjunctive therapy.

An **adjunctive treatment** is a treatment used in conjunction with another to increase the chance of cure or to augment the efficacy of another initial treatment. A new URI for the primary treatment must be introduced to which is associated the adjunctive drug therapy. The indicated drug is linked to this adjunctive therapy, whereas the disease is linked to the primary treatment. A new predicate `loc:associatedWith` establishes the link with the primary treatment.

Treatment	Definition	# occ.
Treatment	No specificity indicated.	1809
Management	Helps the patient to manage a chronic disease.	308
Prophylactic therapy	Used for prevention	175
Symptomatic treatment	Relief of symptoms	112
Topical treatment	Medication applied to the skin surface.	79
Combination therapy	Simultaneous use of a variety of drugs.	53
Short-term treatment	Used on a short periord	52
Control teatment	Control of physiologic process or disease	52
Adjunctive treatment	Used in conjunction with another to increase chance of cure or augment first treatment's efficacy.	47

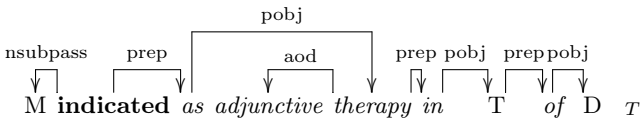
Table 2. Examples of treatments identified from corpus analysis.

Below is an example with its corresponding RDF description.

Entacapone is used as an adjunct to levodopa/carbidopa to treat patients with idiopathic Parkinson's Disease.

```
dailymed_drugs:DB00494  rdf:label "Entacapone".
loc:DT00494A rdf:type loc:DrugTherapy, loc:AdjunctiveTreatment;
             loc:drugInvolved dailymed_drugs:DB00494;
             loc:associatedWith loc:A0023.
loc:A0023 rdf:type loc:Treatment;
          loc:drugInvolved "levodopa/carbidopa";
          loc:target "idiopathic Parkinson's Disease".
```

Pattern



$T \in$ {treatment, prevention, management}

Extraction

$\{U_T$ `rdf:type loc:DrugTherapy`, U_T `rdf:type loc:AdjunctiveTreatment`, U_T `loc:associatedWith` $U_P\} \cup$
$Drug(U_D, U_T, subtree(M)) \cup Treatment(U_P, sub(T)) \cup Disease(U_P, sub(D)) \cup$
$Qualif(U_P, sub(T))$

Fig. 3. Rule Adj_0.

4.3 Refinement of the extraction method

The main challenge in the extraction process is to develop appropriate extraction rules and extraction functions for different types of expressions of the same knowledge. New extraction rules must be defined if the variation affects directly the dependency tree. Otherwise, if the variation occurs within the subtrees, the

extraction functions must be refined. The same two examples, adjunctive therapy and combination therapy, are used to explain this refinement process.

For adjunctive therapy, the variations are at the dependency tree level. Figure 3 shows a new Rule Adj_0 to process sentences such as *Gabapentin is indicated as adjunctive therapy in the treatment of partial seizures.* Note that an additional URI must be provided to represent the primary treatment.

Name	Example	Freq.
Tr_1	Univasc is indicated **for treatment of** patients **with** hypertension. (prepositional phrase attached to *patients*)	111
Tr_2	Univasc is indicated **for treatment of** patients **with** hypertension. (prepositional phrase attached to *treatment*)	140
Tr_3	Camptosar is also indicated **for patients with** metastatic carcinoma of the colon or rectum.	43
Tr_4	Citalopram is indicated **for the treatment of** depression.	2729
Sympt_0	Surmontil is indicated **for the relief of symptoms of** depression.	80
Adj_0	Gabapentin is indicated **as adjunctive therapy in the treatment of** partial seizures.	44
Adj_1	Trihexyphenidyl HCl tablets are indicated **as an adjunct in the treatment of** all forms of parkinsonism.	18
Adj_2	Glipizide tablets are indicated **as an adjunct to** diet **for the control of** hyperglycemia.	77
Cont_0	Provigil is indicated **to improve** wakefulness **in** adult patients with excessive sleepiness associated with narcolepsy	120
Red_0	Inapsine is indicated **to reduce the** incidence **of** nausea and vomiting associated with surgical and diagnostic procedures.	49
Neg_0	Kemstro is **not** indicated in the treatment of skeletal muscle spasm.	237

Table 3. Example sentences for most frequently used rules. Last column indicates the frequency of rule application in our evaluation corpus.

4.4 Final ruleset

Following our corpus analysis, we developed 15 extraction rules grouped in 7 categories. Table 3 gives example sentences for the most frequently used rules. Some rules are mostly variations due to the fact that there are many ways to express the same thing and that syntactic parsers tend to make mistake with prepositional attachment. According to the semantics of the sentence, the phrase *with D* should not be attached to the main verb, but it is frequently interpreted that way by the parser (115 occurrences in our data). We have no choice but to add an additional rule to catch these cases.

The order in which rules are applied is important. Negation is the first pattern that must be tested, since in this case no description must be extracted. For example, when sentence *Flovent Diskus is not indicated for the relief of acute bronchospasm* is processed, it must be recognized that it is a negation and an empty set must be returned. Then rules are applied in order of specificity, with Rule Tr_4 (the most general rule) triggered by default when no other rule applies.

5 Evaluation

We wanted to evaluate the capacities of our system to discover a maximum of drug-disease facts. We also wished to evaluate if the rules actually allow to extract the desired information with accuracy. According to this, our evaluation is intended to measure two criteria, first **coverage** and second **precision**, according to a selected set of predefined rules applied to a reference corpus. To build our experimental protocol, we have manually devised 15 rules (as shown in table 3, inspired by statistics on our corpus (frequent patterns as presented in section 1) and measured how many examples of the reference corpus we correctly found with these rules.

5.1 Coverage evaluation

To get an idea of the overall coverage of our extractor, we selected from Dailymed all sentences that contain the verb *indicated*. This represents 5325 sentences. For 3580 of these sentences (67%), a pattern has been recognized. With this small manual effort (only 15 rules), we cover most of the relevant patterns found in the corpus. We look further to find out what sentences were left out.

First, we find that many sentences are actually irrelevant for our purpose, such as the following ones:

The routes of administration and indicated concentrations for mepivacaine are...
Amiodarone also can be used to treat patients with VT/ VF for whom oral amiodarone is indicated, but who are unable to take oral medication.
Renal function studies should be performed when indicated.

Also, we see that some cases require anaphora resolution, which is outside the scope of our research: *they are also indicated for use in secondary amenorrhea*. But mostly, which might be surprising within such a restricted corpus, the syntactic variations are incredibly high. Consider for example the following sentences, which are not correctly recognized by our patterns:

INDOCIN I.V. is indicated to close a hemodynamically significant patent ductus arteriosus in premature infants weighing between 500 and 1750 g when after 48 hours usual medical management (e.g., fluid restriction, diuretics, digitalis, respiratory support, etc.) is ineffective.
Vinorelbine is indicated as a single agent or in combination with cisplatin for the first-line treatment of ambulatory patients with unresectable, advanced nonsmall cell lung cancer (NSCLC).

Then, within the 3580 sentences covered, we look at the distribution among our rules. Last column of table 3 provides, for each rule of our extractor, the number of times it has been used when processing the whole corpus. Note that the total of rules used is higher than the total of sentences for which a description has been extracted. The reason for this is that, as we have seen, some rules require the recursive application of other rules. An important observation is the necessity of detecting negative forms: the second most used rule is the one that recognizes this form. Also, considering the distribution, we see that even if the default rule is by far the most used (about 77%), the importance of other rules is not negligible. And, as we will see later, there are many cases where Rule Tr_4 was activated erroneously, that is, a specific case has not been recognized by

one of the other rules and should have been. Now, considering rule groups, we see that none seem to be irrelevant, with the exception of rules for combination therapy. In this last case, it appears that almost all instances are detected by the $Drug()$ extraction function (62 occurrences).

5.2 Precision evaluation

To evaluate the precision of our extractor, we manually looked at negative and positive instances. First, we randomly selected a set of 100 sentences for which no description have been generated (negative instances) by our extractor. For each sentence, we determined whether it should have been covered by our extractor (false negatives), that is, whether it does not fall out of the extractor's vocabulary and describes the kind of treatment that is extracted by our rules. We did the same with sentences for which an RDF description has been generated (positive instances): we randomly selected 150 of them and determined, for each one, if the extracted RDF description is correct (true positives). We computed precision (P) and true negative ratio (TNR), using the following formulas (TP = true positives, FP = false positives, TN = true negatives and FN = false negatives):

$$P = \frac{TP}{TP + FP} \qquad TNR = \frac{TN}{TN + FN}$$

Results are given at Table 4. Note that for precision we provide two measures called **Strict** and **Relaxed**. In the strict evaluation, a description is considered correct if it contains all the expected triples. In the relaxed evaluation, we accept partial descriptions. For example, if the sentence refers to a palliative treatment and the extractor returns the description of a generic treatment, it would be accepted. It may also be the case that the drug is indicated for two diseases (denoted by a coordination in the sentence), and only one appears in the description. Table 4b shows the results for each rules individually (values do not sum up to 150 because more than one rule may be applied to the same sentence).

TP	FP	Precision
53	22	0.67 (strict evaluation)
64	11	0.85 (relaxed evaluation)

TN	FN	True negative ratio
79	21	0.79

(a)

Rule group	# occ.	Precision
Tr	122	0.64 (0.83)
Adj	9	0.89 (0.89)
Sympt	8	1,00 (1.00)
Cont	8	0.8 (0.89)
Red	6	0.83 (1.00)
Comb	5	0.80 (1.00)

(b)

Table 4. Evaluation results. (a) Precision and true negative ratio. (b) Precision values for each individual rule group. Values for relaxed evaluation are given within parentheses.

The results show that precision is high for both strict and relaxed measures, with the exception of Tr and Cont rules, where values are low for strict evalu-

ation. It is important to remember that cases rejected by strict evaluation are valid representations. Their only problem is that some information is missing.

6 Conclusion

We presented a syntactic-based method for knowledge extraction. We looked at one type of information, the object literal of the indication predicate in DailyMed dataset. Analysing this data with simple 5-gram frequency analysis, we discovered variations in types of treatment and drug therapy which we captured with variations in our knowledge schema. We developed rules based on dependency trees to extract the required information from the long literal providing this information in natural language text. We showed high precision results on the extracted knowledge.

This present work is in contrast to our previous work investigating shallow textual analysis methods applied to the Web at large, in search of drug indications[17]. The noisy data found was valuable, but that research also showed us that it should be complimentary to data coming from trusted sources. Therefore, in the present research, we look deeper into the specific smaller resource that is the indication predicates of the DailyMed dataset, and see how we can develop precise knowledge extraction methods that will help encode the information within a precise knowledge schema.

Spending the time to refine a RDF structure of a dataset in any domain can be valuable to the Semantic Web community. Especially if such dataset contains information found from trusted sources and is only partially RDFized. The value of the Semantic Web is in its capability of sharing and linking information. This assumes breaking down the information to the notion of concepts (URIs) and labels for such concepts. We see our research as giving a method for analysing textual data, promoting the idea of taking the time to define specialized schema and specialized extraction rules which can then accelerate largely the extraction of knowledge from the dataset.

There are many paths for future work. One is to use the same approach on other predicates in other domains. Another is to go back to knowledge extraction in larger, noisier data, and exploit the benefit of knowledge learned from smaller focused data to help the knowledge extraction process.

References

1. M. Samwald, A. Jentzsch, C. Bouton, C. S. Kallesøe, E. Willighagen, J. Hajagos, M. S. Marshall, E. Prud'hommeaux, O. Hassanzadeh, E. Pichler, and S. Stephens, "Linked open drug data for pharmaceutical research and development," *Journal of Cheminformatics*, vol. 3, no. 19, 2011.
2. P. Buitelaar, P. Cimiano, and B. Magnini, "Ontology Learning from Text : An Overview," in *Ontology Learning from Text: Methods, Evaluation And Applications*, IOS, Ed. IOS Press, 2003, pp. 1–10.

3. S. Brin, "Extracting patterns and relations from the world wide web," *The World Wide Web and Databases*, 1999. [Online]. Available: http://www.springerlink.com/index/446655KM73620362.pdf
4. E. Agichtein, "Extracting relations from large plain-text collections," in *5th ACM International Conference on Digital Libraries*, 2000.
5. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives, "DBpedia: A Nucleus for a Web of Open Data," *In 6th Int'l Semantic Web Conference, Busan, Korea*, pp. 11–15, 2007. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.69.5249
6. M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *of the Joint Conference of the*, no. 2005, 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1690287
7. T. Cassidy, Z. Chen, J. Artiles, H. Ji, H. Deng, L.-a. Ratinov, J. Zheng, J. Han, and D. Roth, "CUNY-UIUC-SRI TAC-KBP2011 Entity Linking System Description," in *TAC-KBP2011*, no. 2, 2011.
8. Z. Chen, S. Tamang, A. Lee, X. Li, W.-p. Lin, M. Snover, J. Artiles, M. Passantino, and H. Ji, "CUNY-BLENDER TAC-KBP2010 Entity Linking and Slot Filling System Description," in *TAC KBP 2010*, 2010.
9. Y. Shinyama and S. Satoshi, "Preemptive information extraction using unrestricted relation discovery," *Proceedings of NAACL HLT*, no. June, pp. 304–311, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1220874
10. P. Pantel, "Espresso: Leveraging generic patterns for automatically harvesting semantic relations," in *Proceedings of Inference in Computational Semantics (ICoS-06)*, no. Hindle 1990, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1220190
11. H. Ji, R. Grishman, H. Dang, and K. Griffitt, "Overview of the TAC 2010 knowledge base population track," *Proc. TAC2010*, 2010. [Online]. Available: http://nlp.cs.qc.cuny.edu/kbp2011.pdf
12. G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel, "The automatic content extraction (ACE) program–tasks, data, and evaluation," in *Proceedings of LREC*, vol. 4. Citeseer, 2004, pp. 837–840.
13. M. Miwa, S. Pyysalo, T. Hara, and J. Tsujii, "A comparative study of syntactic parsers for event extraction," in *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, 2010.
14. J.-D. Kim, Y. Wang, T. Takagi, and A. Yonezawa, "Overview of genia event task in bionlp shared task 2011," in *BioNLP Shared Task 2011, Workshop 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.
15. S. Ananiadou, S. Pyysalo, J. Tsujii, and D. B. Kell, "Event extraction for systems biology by text mining the literature." *Trends in biotechnology*, vol. 28, no. 7, pp. 381–90, Jul. 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20570001
16. F. Rinaldi, G. Schneider, and K. Kaljurand, "Dependency-based relation mining for biomedical literature," in *6th edition of the Language Resources and Evaluation Conference (LREC 2008)*, Marrakech, 2008.
17. C. Barrière and M. Gagnon, "Drugs and disorders: From specialized resources to web data," in *Workshop on Web Scale Knowledge Extraction, 10th International Semantic Web Conference*, 2011.