EquatorNLP: Pattern-based Information Extraction for Disaster Response

Lars Döhling and Ulf Leser

Humboldt-Universität zu Berlin,
Department of Computer Science,
Unter den Linden 6, 10099 Berlin, Germany
{doehling,leser}@informatik.hu-berlin.de

Abstract. One of the most severe problems in early phases of disaster response is the lack of information about the current situation. Such information is indispensable for planning and monitoring rescue operations, but hardly available due to the breakdown of information channels and normal message routes. However, during recent disasters in developed countries, such as the flooding of New Orleans or the earthquake in New Zealand, a wealth of detailed information was posted by affected persons in media, such as Flickr, Twitter, or personal blogs. Finding and extracting such information may provide valuable clues for organizing aid, but currently requires humans to constantly read and analyze these messages. In this work, we report on a study for extracting such facts automatically by using a combination of deep natural language processing and advanced machine learning. Specially, we present an approach that learns patterns in dependency representations of sentences to find textually described facts about human fatalities. Our method achieves a F1 measure of 66.7% on a manually annotated corpus of 109 news articles about earthquake effects, demonstrating the general efficacy of our approach.

Keywords: Information Extraction, Dependency Graph, Earthquake, Disaster Response, Named Entity Recognition, Relationship Extraction

1   Introduction

After disastrous events like earthquakes, decision makers require precise and timely information about the current situation for planning and monitoring rescue operations effectively. During the last years, the Internet has become a major source for such information, in particular, if no acquaintance is available on-site. For earthquake events, many key information like the number affected are published on the Internet. This includes structured information provided by earthquake agencies (e.g. GEOFON[1] or USGS[2]) as well as textual updates published

---

[1] http://geofon.gfz-potsdam.de/geofon
[2] http://earthquake.usgs.gov/earthquakes

by news agencies or recently by Internet users themselves, called user-generated content (e.g. Twitter or personal blogs). Given the example sentence "The death toll in an earthquake in south-west China is now at least 32, with 467 injuries, state media says."[3], one can identify several text snippets expressing presumably demanded facts. It contains trigger words like "death toll" or "earthquake", figures like "32" or "467" as well as temporal ("now") or spatial ("south-west China") attributes. Furthermore, these token or token sequences – subsequently called entities – are semantically connected to each other, forming so called relationships; "death toll" is related to "32" and "at least" whereas "467" refers to "injuries". Moreover, both are associated with "earthquake" and "China". Obviously, texts offer valuable information for decision making but require accurate analysis, which is still a manual and therefore time-consuming, expensive task. Hence, automating this analysis will aid humans to accomplish rescue operations successfully.

As a first step towards automatic textual analysis, we report on extracting facts from news articles, describing human impacts from earthquakes. To model these impacts, we define a 5-ary relationship, whose complexity imposes several challenges for extraction by

- consisting of more than two entities,
- allowing incomplete tuples and
- potentially spanning multiple sentences.

For extracting this relationship, we apply deep natural language processing combined with graph-based synthesis techniques. More specifically, we match patterns in sentence-based dependency graphs to compose a graphical model representing semantic connections between entities and examine this for connected subgraphs. Our evaluation demonstrates the general efficacy of our proposed method stack – called EquatorNLP[4] – by achieving 66.7% F1 measure [23] on a novel, manually created news corpus.

1.1 Related Work

Due to the increasing amount of information available in a textual form (e.g. PubMed or Wikipedia), assisting humans by automatically analyzing these texts has become an important research topic in the last decade. Information extraction (IE) studies the problem of extracting structured information from unstructured text. Typically, this involves recognizing entities (named entity recognition, NER) and relationships between them (relationship extraction, RE).

Different methods have been proposed for NER, e.g. dictionary-based, rule-based or machine learning [20,29]. Hybrids like the one applied in this study usually perform best [28]. The achievable F1 measure highly depends on the concrete domain and ranges up to 95% [10,19,14,32]. To the best of our knowledge,

---

[3] http://news.bbc.co.uk/2/hi/asia-pacific/7591152.stm
[4] *E*arth*QU*ake d*A*ta collec*TOR* [8] with *N*atural *L*anguage *P*rocessing

this study is the first about IE in the earthquake domain, hence no quantitative results are available yet.

Regarding RE, co-occurrence forms an intuitive approach [15]. Beside that, pattern matching [30] and machine learning [21] have been adopted as well. As in EquatorNLP, these methods recently utilize deep natural language processing like dependency parsing [12]. Little is known about extracting complex *n*-ary relationships like the one examined in this paper, since most research has focused on binary relationships. Inspired by the promising results in [27], we transferred their subgraph-based idea into our domain (see section 2.4). In general, RE is regarded as being more difficult than NER, resulting in lower F1 measures, ranging from 40% [16] to 80% [12].

2 Materials and Methods

2.1 What we extract: Definition of the 5-ary Relationship

To model earthquake damages, our examined relationship consists of five different entity types, including several subtypes. Note that the concatenated parenthesized letters will subsequently be used as abbreviations.

- (O)bject: Describes the victims, e.g. "people" or "students".
- (Q)uantity: Describes the number of victims and consists of the four subtypes
  - (c)ardinal: "12", "ten", "no", "a", "1.3 million"
  - (o)rdinal: "second", "10th"
  - (v)ague: "many", "hundreds", "some"
  - (r)esidue: "everybody"
- (M)odifier: Refers to a quantity and modifies its value, e.g. "at least", "about" or "more than".
- (I)ndicator: Describes the type of damage and consists of six subtypes
  - (k)illed: "killed", "death tool", "died"
  - (i)njured: "injured"
  - (t)rapped: "trapped"
  - (m)issing: "missing"
  - (h)omeless: "homeless"
  - (a)ffected: "affected"
- (N)egation: Infrequently required to correctly describe a damage, e.g. "not".

Given this definition, the previous example "The death toll in an earthquake in south-west China is now at least 32, with 467 injuries, state media says." contains five entities: "death toll" (Ik), "at least" (M), "32" (Qc), "467" (Qc) and "injuries" (Ii). Note that entities may span multiple token – called multi-token entities. Together, these entities form two [N, M, Q, O, S] relationship tuples: [—, "at least", "32", —, "death toll"] and [—, —, "467", —, "injuries"]. We define that not all entity slots have to be filled to form a valid tuple, indicated by —. However, we postulate two constraints concerning incomplete relationship instances: (i) An entity I is mandatory and (ii) an entity Q is mandatory, if an entity M is set.

2.2 Corpus

To train and later test our proposed machine-learning-based extraction methods, we required an annotated set of documents – called corpus – as a gold standard. As to the best of our knowledge, currently no appropriate corpus exits for our purpose, we created a new one. Our corpus consists of 109 English articles about earthquakes and their aftermath: 24 from BBC News[5], 2 from Equator [8], 41 from Wikipedia[6] and 42 from Yahoo! News[7]. They were randomly selected from a collection of documents retrieved from these four sources in spring 2010.

From each article, we extracted the text including the headline and annotated it manually according to the relationship definition given above. We removed cross-sentence (28) and unary (4) instances from the corpus, since our relationship extraction methods operate on the sentence level and are unsuitable for unary tuples (see section 2.4). Finally, we partitioned this altered corpus into a training ($\frac{2}{3}$) and an evaluation set ($\frac{1}{3}$) by stratified random sampling on the sentence level. Table 1 presents the resulting distribution of the relationship tuples in the different partitions.

Table 1. Data set statistics; Note that the Gold Standard values differ from the sum of training and evaluation set, owing to the removal of unary and cross-sentence instances.

Number of [...]		Training	Evaluation	Gold Standard
Sentences		1,964	986	2,950
containing a relationship instance		276	145	486
Token		39,856	20,796	60,652
Relationship instance		382	190	604
per type, defined by I sybtype	k	273	135	439
	i	56	24	80
	t	15	7	23
	m	19	11	30
	h	17	10	27
	a	2	3	5
per size, defined by filled entity slots	1	0	0	4
	2	152	69	245
	3	156	76	236
	4	74	45	119
	5	0	0	0

2.3 Named Entity Recognition

A prerequisite for relationship extraction is the detection of target entities in the text. For this task, we used a regular expression (`Qc` only) in combination with a

---

[5] `http://news.bbc.co.uk`

[6] `http://en.wikipedia.org/wiki/Historical_earthquakes`,
  `http://en.wikipedia.org/wiki/List_of_20th_century_earthquakes`,
  `http://en.wikipedia.org/wiki/List_of_21st_century_earthquakes`

[7] `http://news.yahoo.com/science/earthquakes`

dictionary (all other types), both derived from the training data. As each token sequence can only be assigned to at most one entity type, the question emerged how to disambiguate competing matches. we applied the following plausible order of precedence: The regular expression matches prior to the dictionary, longer token sequences match prior to shorter ("as high as" M versus "high" Qv) and finally the most frequent type found for this token sequence in the training data. Overall, the dictionary extracted from the training data set contained 218 entries with an average length of 1.78 token.

2.4 Relationship Extraction

After recognizing the entities, the next step is to extract the actual relationship instances. Our proposed method consisted of two steps:

1. Discovering pairs of entities by pattern matching in dependency graphs.
2. Synthesizing complex instances from maximal cliques in entity graphs build from these entity pairs.

Dividing the extraction process into these two steps enabled us to apply well-known extraction methods for binary relationships. Furthermore, we gained more training instances, reducing the sparse data problem [22] existing for the complete relationship.

Matching Dependency Patterns Dependency models are syntactical models expressing the hierarchical dependencies between the words of a sentence. Those dependencies may be visualized as a directed, labeled graph whose root is the verb. Figure 1 depicts the running example in the Stanford Dependencies representation [25]. The arrows indicate the dependency direction from regent to dependent and are labeled with the dependency type.

These models offers a direct access to sentence structures [23] and have the potential to reveal relations between words apart more easily than regular expressions [12,7] (e.g. between "death toll" and "32" in the example). Therefore, examining patterns between entities in dependency graphs has been a successful approach in modern relationship extraction. For our work, we selected the shortest paths between two entities as patterns [4].

We applied the Stanford converter [24] to compute the dependency graph for each sentence, which requires constituent parses [23,1] as input (another syntactical model). Those parses were generated by the Charniak PCFG parser [5] in combination with the Charniak-Johnson Max-Ent reranking parser [6], using McClosky's self-trained models [26].

During training, we extracted all shortest paths between entities and stored them in a pattern catalog. To abstract from the actual token of an entity, we joined all parting token vertices in advance into one entity vertex. Concurrently, we replaced each entity vertex in the pattern by its type to mask the actual value. Figure 2 illustrates the transformed example graph and the extracted patterns. Since dependency graphs can contain cycles, there might exist more than one
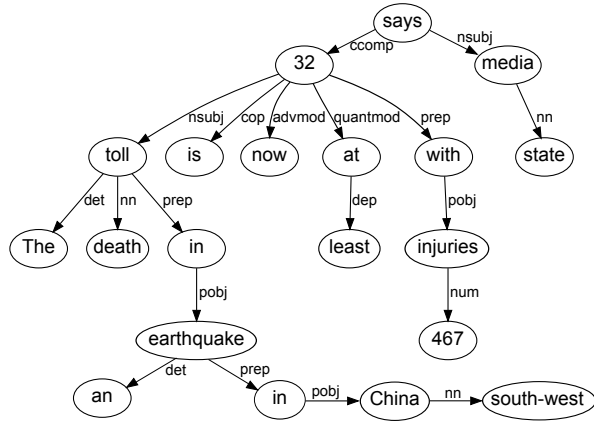
Fig. 1. Dependency graph of the sentence "The death toll in an earthquake in south-west China is now at least 32, with 467 injuries, state media says."

shortest path between two entities with – of course – equal length. Hence, a relationship instance consisting of k entities will produce at least $\binom{k}{2}$ patterns. Overall, the catalog extracted from the training data set contained 396 unique patterns with an average length of 2.83 edges.

During extraction, we applied this catalog to create links between two entities in accordingly transformed dependency graphs, resulting in entity graphs (see Figure 3 for an example).
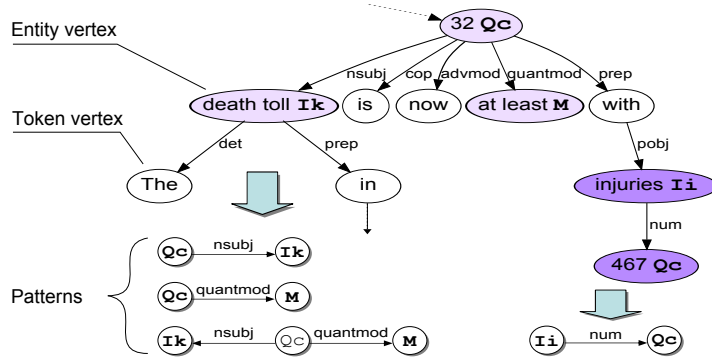

Fig. 2. Pattern extraction in the transformed example dependency graph (truncated)

Baseline To determine whether deep linguistic parsing like the dependency model is beneficial for relation extraction or not, we also use a co-occurrence-based classifier as a baseline for recognizing entity pairs. For each entity e, all

closest (in terms of token distance) entities within sentence scope having a different type then e are postulated as being linked to e. For example, this would imply a (false) connection between "32" and "injuries" in the running example, as for "32" the distance to "injuries" is less then to "death toll".

Synthesizing Relationship Instances After detecting pairs of entities, the final step is to synthesize relationship instances from them. To address this, we identified maximal cliques in the entity graphs [27] which are consistent to our relationship definition.

Consider the entity graph in Figure 3 as one possible outcome of the previous pair-recognizing step when applied to the running example. To form relationship instances, we combined all those entities which are directly connected among each other in the entity graph. Such a set of vertices is called a clique. In Figure 3, all cliques of size two or greater are marked by eclipses (C_0 to C_5). Among these, we considered only those cliques that are consistent to our relationship definition (C_0, C_2 and C_4). Furthermore, we ignored cliques which are contained in others (C_2). All such non-redundant cliques are called maximal. In our example, only C_0 and C_4 comply with the requirements 'maximal' and 'relationship-definition-consistent' and would in this case form the final output of the complete information extraction pipeline.
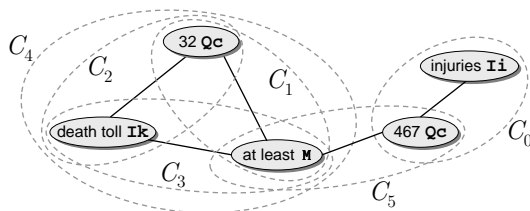
Fig. 3. An entity graph for the example sentence

3   Evaluation and Results

Based on the training data set, we derived optimal extraction pipeline configurations and tested them on the evaluation set. Before presenting our findings, we will explain the evaluation measures used and the underlying configuration parameters.

3.1   Evaluation Measures

To measure the performance of our pipeline, we determined precision (P), recall (R) and F1 measure [23] for all three extraction steps: recognizing entities (NER), extracting entity pairs (BinRE) and synthesizing relationship instances (RE).

Each measure is based on the concept of 'true positive'. We applied a strict evaluation schema, therefore considering a reported entity as a true positive, if and only if both the type and the token agreed with the gold standard [20]. Propagated to the relationship level, an instance was considered a true positive if and only if all participating entities were true positives and the instance had equal size.

3.2 Pipeline Configuration Parameters

Both NER and dependency-based BinRE are requiring well-defined matching criteria. On the entity level, we applied character-based equality. This could be relaxed by case insensitivity (IgnoreCase4NER) or stemming [31,17,23] (UseStem4NER). On the dependency level, we chose between different dependency schemata [25] (DependencySchema). Furthermore, we altered the token vertex matching by case insensitivity (IgnoreCase4RE), stemming (UseStem4RE) or using Part-Of-Speech tags [11,23] (UsePOS4RE). For matching entity vertices, we additionally ignored the subtype (IgnoreEntitySubtype). Moreover, matching pattern edges was modified by ignoring their direction (IgnoreDepDirection) and their label (IgnoreDepType). Given these parameters, Table 2 lists the configurations for maximal precision, recall and F1, estimated from stratified 10-fold cross-validation [13] on the training data.

Table 2. Optimal matching configurations; active: $+$, inactive: $-$

Parameter	P_{max}	R_{max}	$F1_{max}$	OracleNER $F1_{max}$
IgnoreCase4NER	$+$	$+$	$+$	
UseStem4NER	$-$	$+$	$+$	
DependencySchema	CollapsedTree	CCprocessed	Collapsed	CCprocessed
IgnoreCase4RE	$-$	$-$	$-$	$-$
UseStem4RE	$+$	$-$	$+$	$-$
UsePOS4RE	$-$	$+$	$-$	$+$
IgnoreEntitySubtype	$+$	$+$	$+$	$+$
IgnoreDepDirection	$-$	$+$	$-$	$+$
IgnoreDepType	$-$	$+$	$-$	$+$

3.3 Results

Based on the previously deduced pipeline parameters, we evaluated our proposed methods on the evaluation data set. The results for each pipeline step are shown in Table 3. While our approach achieved a surprisingly high recall for entity recognition (93.8%), the corresponding precision was quite low (22.7%). Further analysis revealed that the majority of false positives were produced by the regular expression matching each number in the text (e. g. year or monetary amount).

On the entity pair level, our proposed dependency pattern matching significantly outperformed the baseline in terms of precision (73.0% versus 29.0%). Considering recall, the relation was inverted (74.3% versus 87.2%), resulting in a

Table 3. Evaluation results for different pipeline setups, supplemented by boot-strapped 95% BC_α confidence intervals [9]

Pipeline Setup	NER			BinRE			RE		
	P	R	F1	P	R	F1	P	R	F1
Baseline	.227	.938	.366	.290	.872	.436	.260	.637	.369
	+.033	+.018	+.042	+.039	+.035	+.044	+.038	+.068	+.046
	−.032	−.022	−.043	−.038	−.044	−.046	−.035	−.076	−.045
Dependency P_{max}	.219	.942	.355	.748	.727	.737	.783	.568	.659
	+.032	+.018	+.042	+.052	+.059	+.046	+.062	+.073	+.061
	−.031	−.022	−.042	−.061	−.067	−.052	−.077	−.078	−.069
R_{max}	.207	.948	.339	.553	.836	.666	.403	.711	.514
	+.031	+.017	+.041	+.046	+.045	+.039	+.052	+.066	+.051
	−.029	−.022	−.041	−.049	−.059	−.042	−.051	−.078	−.053
$F1_{max}$	.207	.948	.339	.730	.743	.736	.767	.589	.667
	+.031	+.017	+.041	+.053	+.058	+.045	+.062	+.072	+.058
	−.029	−.022	−.041	−.061	−.066	−.050	−.076	−.079	−.069
OracleNER & Baseline				.867	.984	.922	.743	.884	.808
				+.040	+.012	+.026	+.072	+.049	+.060
				−.056	−.028	−.043	−.093	−.077	−.086
& Dependency $F1_{max}$				.930	.906	.918	.813	.826	.820
				+.031	+.036	+.026	+.070	+.059	+.054
				−.057	−.053	−.037	−.124	−.079	−.083

significantly higher F1 measure for the former (73.6% versus 43.6%). Obviously, matching dependency patterns is more insusceptible to low-precision NER than co-occurrence-based classification.

The same tendencies were observed for relationship instances with a significantly better F1 measure of 66.7% versus 36.9%. Additional examination showed that, for both methods, the reported overall precision and recall scores were roughly consistent across instance types (k, i...) and sizes (2, 3...).

Due to EquatorNLP's pipeline architecture, the observed BinRE and RE performances were certainly biased by the preceding NER step. To quantify the effect of error propagation and therefore disclosing their 'true' capabilities, we also tested a perfect NER (OracleNER in Table 2 and 3). Although our results confirmed the global trends for the distribution of precision, recall among the two BinRE methods, their absolute difference in F1 measure were nearly eliminated (82.0% versus 80.8%).

4 Conclusions and Future Work

In this paper, we demonstrated that matching dependency patterns combined with detecting maximal cliques is a promising approach for extracting human

impacts from earthquake reports. Our evaluation on a manual annotated corpus resulted in a maximal F1 measure of 66.7%, outperforming a co-occurrence-based approach significantly. We also showed that our proposed extraction pipeline provides P / R adaptability. Additional experiments with oracle NER imply that under this setting, co-occurrence-based extraction provides competitive results, particularly with regard to its significantly lower computational runtime [2].

Note that the computed recall and F1 measures are slightly biased, since our proposed extraction pipeline operates only on the sentence level and does not cover unary relationship instances. As these instances form approximately 5% of all tuples in news articles (see section 2.2), this might be acceptable for particular applications.

As stated before, our evaluation was focused exclusively on domain specific texts. We cannot expect the same performance for unfiltered texts. The application on 113 general news articles yielded a precision of only 3.2%. This result is less surprising if one takes a closer look at Figure1, showing that the domain trigger "earthquake" is not part of any shortest path between entities. In fact, only 1 out of all 396 extracted patterns contains a trigger word. Certainly, incorporating semantic knowledge for filtering texts would increase precision. On the other hand, this nonspecifity might be considered as an advantage, suggesting that our pipeline is applicable to other types of disasters.

To finally set the achieved F1 measure of 66.7% in context to a prospective human performance, we assessed this by calculating the inter annotator agreement (IAA) [3] for two independent annotations. The score of 70.3% for strict agreement on relationship instances for 30 articles indicates at least a cardinal task complexity. Great caution should be exercised in comparing these two values directly, since they belong to different dimensions: the former measures validity, while the latter measures objectivity.

4.1 Future Work

Given these results and our conclusions, we identified several challenges for future research. Obviously, we require domain specific texts as pipeline input, proposing text classification as a preprocessing step. As decision makers are interested in information about specific events, we plan to extend our relationship and its extraction to temporal and spatial attributes. Furthermore, we intend to apply high-precision machine learning techniques like condition random fields [18] for NER, hopefully increasing RE recall without losing precision by enabling less strict pattern matching criteria. Finally, we intend to explore user-generated content like on Twitter as a novel information source.

References

1. Bies, A., Ferguson, M., Katz, K., MacIntyre, R., Tredinnick, V., Kim, G., Marcinkiewicz, M.A., Schasberger, B.: Bracketing Guidelines for Treebank II Style, Penn Treebank Project (1995)
2. Bjorne, J., Ginter, F., Pyysalo, S., Tsujii, J., Salakoski, T.: Complex event extraction at pubmed scale. Bioinformatics 26(12), 382–390 (June 2010)
3. Brants, T.: Inter-annotator agreement for a german newspaper corpus. In: Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000) (2000)
4. Bunescu, R., Mooney, R.: A shortest path dependency kernel for relation extraction. In: HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. pp. 724–731. Association for Computational Linguistics, Morristown, NJ, USA (2005)
5. Charniak, E.: A maximum-entropy-inspired parser. In: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference. pp. 132–139. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2000)
6. Charniak, E., Johnson, M.: Coarse-to-fine n-best parsing and maxent discriminative reranking. In: ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 173–180. Association for Computational Linguistics, Morristown, NJ, USA (2005)
7. Clegg, A., Shepherd, A.: Benchmarking natural-language parsers for biological applications using dependency graphs. BMC Bioinformatics 8(1), 24+ (2007)
8. Döhling, L., Woith, H., Fahland, D., Leser, U.: Equator: Faster decision making for geoscientists. In: Proceeding of Workshop on IT support for rescue teams 2011 (2011), (to appear)
9. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall/CRC (1993)
10. Farmakiotou, D., Karkaletsis, V., Koutsias, J., Sigletos, G., Spyropoulos, C.D., Stamatopoulos, P.: Rule-based named entity recognition for greek financial texts. In: In Proceedings of the Workshop on Computational Lexicography and Multimedia Dictionaries (COMLEX 2000). pp. 75–78 (2000)
11. Francis, W.N., Kucera, H.: Brown Corpus Manual (1979)
12. Fundel, K., Küffner, R., Zimmer, R.: RelEx – Relation extraction using dependency parse trees. Bioinformatics 23(3), 365–371 (2007)
13. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, 2nd ed. edn. (4 2006)
14. Isozaki, H., Kazawa, H.: Efficient support vector classifiers for named entity recognition. In: Proceedings of the 19th international conference on Computational linguistics. pp. 1–7. Association for Computational Linguistics, Morristown, NJ, USA (2002)
15. Jenssen, T.K., greid, A.L., Komorowski, J., Hovig, E.: A literature network of human genes for high-throughput analysis of gene expression. Nature genetics 28(1), 21–28 (5 2001)
16. Kawtrakul, A., Yingsaeree, C., Andrès, F.: A framework of nlp based information tracking and related knowledge organizing with topic maps. In: NLDB. pp. 272–283 (2007)
17. Kraaij, W., Pohlmann, R.: Viewing stemming as recall enhancement. In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 40–48. SIGIR '96, ACM, New York, NY, USA (1996)

18. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 282–289. Morgan Kaufmann (2001)

19. Leaman, R., Gonzalez, G.: Banner: an executable survey of advances in biomedical named entity recognition. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing pp. 652–663 (2008)

20. Leser, U., Hakenberg, J.: What makes a gene name? named entity recognition in the biomedical literature. Briefings in Bioinformatics 6(4), 357–369 (2005)

21. Li, J., Zhang, Z., Li, X., Hsinchun, C.: Kernel-based learning for biomedical relation extraction. Journal of the American Society for Information Science and Technology 59(5), 756–769 (2008)

22. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (7 2008)

23. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, 2nd printing w. corrections edn. (6 2000)

24. Marneffe, M., MacCartney, B., Manning, C.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC-06. pp. 449–454 (2006)

25. de Marneffe, M.C., Manning, C.D.: Stanford typed dependencies manual, revised in february 2010 edn. (2008)

26. McClosky, D., Charniak, E., Johnson, M.: Effective self-training for parsing. In: Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. pp. 152–159. Association for Computational Linguistics, Morristown, NJ, USA (2006)

27. McDonald, R., Pereira, F., Kulick, S., Winters, S., Jin, Y., White, P.: Simple algorithms for complex relation extraction with applications to biomedical ie. In: ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 491–498. Association for Computational Linguistics, Morristown, NJ, USA (2005)

28. Mikheev, A., Moens, M., Grover, C.: Named entity recognition without gazetteers. In: Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics. pp. 1–8. Association for Computational Linguistics, Morristown, NJ, USA (1999)

29. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Linguisticae Investigationes 30(1), 3–26 (January 2007)

30. Pietschmann, S.: Relationship extraction by frequent patterns in dependency graphs (in German). Diplom thesis, HU-Berlin (September 2009)

31. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (July 1980)

32. Zhou, G., Su, J.: Named entity recognition using an hmm-based chunk tagger. In: ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. pp. 473–480. Association for Computational Linguistics, Morristown, NJ, USA (2002)