Investigating Approaches to Semantic Category Disambiguation Using Large Lexical Resources and Approximate String Matching

Pontus Stenetorp,[†1] Sampo Pyysalo,[†2,†3]
Sophia Ananiadou[†2,†3] and Jun'ichi Tsujii [†4]

This paper proposes and investigates several improvements for an existing machine learning-based system for the task of semantic category disambiguation. For querying large scale lexical resources with millions of lexical entries using approximate string matching we investigate the application of a semantically motivated distance measure, using start/end markers for the query, selecting the most beneficial lexical resources out of a set and the effect using similarity thresholds. These approaches are evaluated using six datasets from the domain of BioNLP and while some modest improvements are observed we fail to establish a consistent benefit for any of the suggested methods for all datasets. The introduced system and all related resources are freely available for research purposes at: `https://github.com/ninjin/simsem`

1. Introduction

Semantic category disambiguation is the task of determining the semantic category (or categories) that a textual span carries in a given context. The task represents a sub-problem for several well-established tasks in Natural Language Processing (NLP), most prominently Named Entity Recognition which conceptually can be seen as two sub-tasks, determining which textual spans that are likely to carry entity mentions and then disambiguating between the possible semantic categories for entities.

Other NLP tasks that semantic category disambiguation has previously been applied to include co-reference resolution[22], where co-referring mentions must share the same semantic category and one can thus exclude unlikely candidates. The added semantic information has been used for the task of coordination analysis[18]; consider the coordinate clause: "Coffee or tea and a sandwich"; knowing that the first two members of the clause are drinks makes it possible to infer that the choice is most likely between which drink you are having with your sandwich, rather than [coffee] or [tea and a sandwich].

This study focuses in particular on the domain of BioNLP where early annotation efforts has covered a multitude of semantic categories[13]. But despite the access to annotated resources with several semantic categories, NER studies have only been concerned with single semantic categories such as "protein" or "species"[6),24]. Over the last few years the field has moved towards extracting events with one or multiple participants, these participants come from multiple semantic categories and has thus put emphasis on the necessity of multi-category NER. Lately, as a part of the BioNLP 2011 Shared Task[9] the participants of the annotated events came from as many as 17 semantic categories. High quality multi-category NER systems have thus become an essential underpinning for applying and forwarding the state-of-the-art for complex event extraction systems in the field of BioNLP.

1.1 Previous Work

In order to boost NER performance state-of-the art NER systems[16),24] utilises lexical resources to serve as a prior and to filter our semantic categories which are easily confused with the target categories ("chemicals" for example follow similar lexical patterns as "proteins"). We previously introduced a system, SimSem[20], that utilised large scale lexical resources for the task of semantic category disambiguation.

The novelty of the system was primarily that it utilised approximate as opposed to strict string matching when performing look-ups for the lexical resources and that the scale of the lexical resources utilised surpassed those of earlier NER systems. The method was evaluated for several datasets with results for macro-level accuracy ranging from 85.9% to 95.3%. However, the study failed to establish a clear systematic benefit of approximate, as opposed to strict, string matching for all datasets and was thus inconclusive. For this study we seek to investigate some possible adjustments to our previously introduced method to better understand the implications of some of the design decisions from our previous study.

†1 Aizawa Laboratory, Department of Computer Science, University of Tokyo, Tokyo, Japan
†2 School of Computer Science, University of Manchester, Manchester, UK
†3 National Centre for Text Mining, University of Manchester, Machester, UK
†4 Microsoft Research Asia, Beijing, People's Republic of China

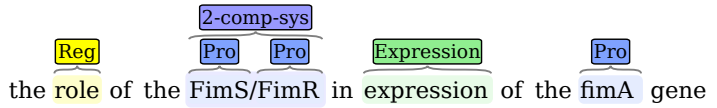

Fig. 1 Typed text-bound annotations[15] of an abstract[28] from the biomedical domain

2. Methods

2.1 Task Setting

Our task setting is to classify a continuous textual span as belonging to one out of several given semantic categories from a fixed set. **Figure 1**⋆1 illustrates our task setting and also the possibility of overlapping spans with different semantic categories.

2.2 Baseline

Since we build upon our previously introduced system[20], this section gives a brief summary of what serves as the baseline for this work. The feature set used is derived from common features used for NER (**Table 1**), but in addition we used SimString⋆2, which is an implementation of the CPMerge algorithm[14], to generate approximate string matching features for the textual span. This was done using 170 databases generated from 10 different lexical resource collections (**Table 2**) containing over 20,000,000 lexical entries. The cosine measure for the tri-grams are used to determine the similarity between two given strings. Using cosine similarity the system has a threshold indicating the percentage of tri-grams that must be shared between the compared strings to consider them as a match. Do note that a match for a threshold of 1.0 indicates that the strings are identical, the same as when using a strict string matching. When querying each resource the system uses a sliding threshold ranging from 1.0 to 0.7, with intermediate steps of 0.1. If a match was found for a given threshold, a binary feature is generated for the given threshold and all lower thresholds for that database since a match for a higher threshold ensures that any lower threshold would also return a match.

⋆1 Visualised using the stav visualiser[21], `https://github.com/TsujiiLaboratory/stav`
⋆2 Version 1.0, `http://www.chokkan.org/software/simstring/`

Table 1 String internal features

Feature	Type	Input	Value(s)
Text	Text	Flu	Flu
Lower-cased	Text	DNA	dna
Prefixes: lengths $[3, 5]$	Text	bull	bul, ...
Suffixes: lengths $[3, 5]$	Text	bull	ull, ...
Stem	Text	performing	perform
Is a pair of digits	Bool	42	True
Is four digits	Bool	4711	True
Letters and digits	Bool	C4	True
Digits and hyphens	Bool	9-12	True
Digits and slashes	Bool	1/2	True
Digits and colons	Bool	3,1	True
Digits and dots	Bool	3.14	True
Upper-case and dots	Bool	M.C.	True
Initial upper-case	Bool	Pigeon	True
Only upper-case	Bool	PMID	True
Only lower-case	Bool	pure	True
Only digits	Bool	131072	True
Only non-alpha-num	Bool	#*$!	True
Contains upper-case	Bool	gAwn	True
Contains lower-case	Bool	After	True
Contains digits	Bool	B52	True
Contains non-alpha-num	Bool	B52;s	True
Date regular expression	Bool	2012-12-20	True
Pattern	Text	1B-zz	0A-aa
Collapsed Pattern	Text	1B-zz	0A-a

2.3 Linguistically Motivated Metrics

While using a cosine measure for approximate string matching enables fast look-ups for strings, it is not linguistically motivated since it assigns equal weight to any n-gram occurring in the string, regardless of the characters and relative position in the string. To address this issue we propose to use a linguistically motivated edit distance such as the Levenshtein distance.

For an edit distance with a fixed cost of 100 “EGR-1”, “EGR 1” and “FGR-1” would all be at the same distance of 100 from each other since they all differ by one character. But as illustrated in **Table 3**, using a variable cost edit distance it is possible to reduce the cost of substituting a hyphen into a space to be lower since it has less significant semantic implications than changing a letter into another letter. This leads to “EGR-1” having a shorter distance to “EGR

Table 2 Lexical resources.

Name	Lexical Entries
Arizona Disease Corpus[4]	1,195
Entrez Gene[11]	3,602,757
Gene Ontology[1]	128'955
Generated dictionary[19]	61,676
Jochem[7]	1,715,744
LINNAEUS Dictionary[6]	2,880,878
Protein Information Resource[27] (PIR)	686,203
Turku Event Corpus[2]	6,273,576
Unified Medical Language System[3] (UMLS)	5,872,202
Websters Second International Dictionary	235,882

Table 3 Symmetric distance-matrix illustrating fixed and variable-cost edit distance for proteins (FIXED/VARIABLE)

	EGR-1	EGR 1	FGR-1
EGR-1	-	100/10	100/100
EGR 1		-	100/100
FGR-1			-

1" than "FGR-1".

We implemented a variable cost edit distance designed to match protein mentions[26]. Since proteins are the dominating category for a majority of our datasets and we should expect to see significant improvements by adopting a measure designed to match proteins. Henceforth, we refer to this measure as EDIT and NEDIT which is normalised in relation to the maximum distance between the two strings.

We sorted the results returned by SimString by cosine similarity to the query string and then applied the edit distance to the top results, making a cut-off at the tenth result from the top. We then used the best match among the distances for our features. Lastly, the distances were bucketed since both EDIT and NEDIT are continuous measures. EDIT use buckets from 0 to 100 with step 10, from 100 to 1000 with step 50 and from 1000 to 10000 with step 1000, followed by a catch-all bucket for all larger numbers. For NEDIT we use buckets ranging from 0.0 to 1.0 with a bucket step size of 0.1. We also applied cascading to address feature sparsity, thus if the feature fired for a given bucket it would also fire for each bucket which would correspond to a worse match than the one it originally fired for.

While our choice of edit distance[26] takes specific characters into consideration it does not consider the location of the characters in the string. For example: any substitution in a string is solely based on the character being substituted and not on the position of the character. A feature of SimString is the possibility to mark the beginning and ending of a string with "guards" and use it for the queries. This is a common feature in NER and since this gives additional information about the structure of the string we chose to investigate the effects of adding guards in addition to our variable cost edit distance and we refer to this feature as GUARDED.

2.4 Resource Selection

We speculated that among the lexical resources there could be some resources that were not beneficial or possibly even harmful to the performance of the classifier. In order to discover which resources provided the least amount of leverage we performed a greedy descent search and iteratively removed resources that were deemed not be beneficial.

2.5 Threshold Tuning

Our previous system[20] used a lower-bound threshold cut-off below which they would not search for any further matches. We considered the possibility that even looser matching could be beneficial. To evaluate this we searched for an optimal threshold by gradually decreasing the threshold from 1.0 to 0.1 by steps of 0.1 and observed if there were any significant changes in performance.

2.6 Metrics

To produce our metrics we gradually increase the amount of training data, starting from 5% to 100% with steps of 5%. For each sampling point we train multiple models using random samples of the training set and use the mean to represent a given data point. We use instance-level accuracy and use the mean of the data points (analogous to the Area Under the Curve) to summarise our performance.

2.7 Models

Based on our feature suggestions we construct 8 different candidate models to compare to our baseline. All models incorporate the string internal (non-

contextual) features (INTERNAL) from Table 1 and we generate the SimString features using the lexical resources in Table 2. As our machine learning component we use LIBLINEAR[⋆1 5)] with a L2-regularised logistic regression model and we optimise the penalty parameter using cross-validation on the training data. This makes our setting identical to that used to evaluate our baseline model from our previous study[20)] and thus makes our results comparable.

We generate models for our proposed linguistically motivated EDIT and NEDIT features, with and without start/end-of string guards (GUARDED).

For the resource selection we used the training portion for each dataset and performed 5-fold cross-validation for each resource and determined if removing a given resource would increase performance. Since this approach is computationally expensive we inspected the results from the cross-validation to see how many iterations each dataset would require until no further significant improvements could be made. After establishing the number of iterations on the training portion of our datasets we combined the training and development sets to determine the least beneficial resources for each dataset, which were then left out when constructing our final RESOURCESELECTION model.

2.8 Corpora

Table 4 describes the corpora used for evaluation. These are the same as in our previous publication[20)] which make comparisons to the baseline straight forward. The datasets are separated into training, development and test sets consisting of 1/2, 1/4 and 1/4. This is done randomly and on an annotation level. The test set was used to generate the final results prior to submitting the publication and was thus not used during development in order to ensure the validity of the final results.

3. Results and Discussion

4. Experimental Set-up

If we refer to **Table 5** which summarises our model suggestions, we can see that the EDIT and NEDIT features fail to be beneficial for any dataset. Introducing guards gave mixed results, but the results differ depending on the cosine threshold being 0.4 or 0.7. For 0.7 the results are overwhelmingly negative, while for 0.4 they are slightly beneficial. This may strike us as odd, but the guards have the side-effect of conceptually raising the cosine threshold since every query now contain two additional characters, the start and end guard. The guards also add additional weight to the start and end of the query. In a practical sense this will result in actions such as capitalisation of the first character and pluralisation and conjugation receiving additional weight relative to the query length.

To find an optimal threshold using our development set we gradually decreased the SimString cosine threshold for our baseline model from 1.0 to 0.1 by steps of 0.1. For each step we observed the effect on the accuracy of the model and found that a threshold of 0.4 greatly outperformed one of 0.7. However, lowering the threshold further than 0.4 did not give any significant improvements in accuracy and we used 0.4 as the threshold for our model suggestions. The results for our new thresholds indicate that there are performance gains to be found; in particular this can be observed for the ID and EPI datasets but the threshold tuning can also incur performance penalties as can be seen for NLPBA among others. In retrospect after taking the results from the guards into consideration, the results suggest that when searching for an optimal threshold the preferable way to do so is in combination with determining if using guards are beneficial.

For the resource selection we found that five iterations were a reasonable cut-off for our greedy descent search. The method shows some promise for GE, SGREC and in particular the ID dataset with 9.8 in error reduction. However, it is clear that there can be very negative implications and that our simple descent can over-fit to the data and have a negative impact such as can be observed with a 11.5 relative error increase for the SSC dataset.

5. Conclusions and Future Research

We have proposed several additions to an existing system and seen indications that performance can potentially be improved by utilising even looser string matching than what was previously proposed. For linguistically motivated measures we have found that adding guards to the beginning and end of strings can be beneficial but are highly sensitive to the threshold used for the string match-

⋆1 Version 1.7 of the software

Table 4 Corpora used for evaluation, the parenthesised value for GREC is for the collapsed superset SGREC which does not suffer from the same level of data sparseness. For the corpora containing event triggers (EPI, ID, GE, GREC and SGREC), the triggers are treated as distinct semantic categories

Name	Semantic Categories
BioNLP/NLPBA 2004 Shared Task Corpus[8] (NLPBA)	5
Gene Regulation Event Corpus[23] (GREC)	64 (6)
Collaborative Annotation of a Large Biomedical Corpus[17] (SSC)	4
Epigenetics and Post-Translational Modifications[12] (EPI)	17
Infectious Diseases Corpus[15] (ID)	16
Genia Event Corpus[10] (GE)	11

Table 5 Learning curve means for our proposed models. Abbreviations used: Int. for Internal, Sim. for SimString, g for Guarded, t for cosine threshold, r for ResourceSelection

Classifier		EPI	ID	GE	SSC	NLPBA	SGREC	μ
Int.[20]		92.5	91.2	**94.6**	81.7	**92.1**	82.1	89.0
Int.Sim.[20]	(t=0.7)	93.7/+16.0	91.8/+6.8	94.4/-3.7	92.2/+57.4	**92.1/0.0**	83.4/+7.3	91.3/+20.9
Int.Sim.	(g,t=0.7)	93.7/0.0	91.7/-1.2	94.5/+1.8	91.0/-15.4	91.9/-2.5	82.9/-3.0	91.0/-3.4
Int.Sim.Edit	(t=0.7)	93.4/-4.8	91.2/-7.3	93.7/-12.5	91.8/-5.1	91.6/-6.3	82.7/-4.2	90.7/-6.9
Int.Sim.Edit	(g,t=0.7)	93.5/-3.2	90.5/-15.9	93.8/-10.7	91.3/-11.5	91.6/-6.3	81.8/-9.6	90.4/-10.3
Int.Sim.NEdit	(t=0.7)	93.5/-3.2	91.2/-7.3	94.0/-7.1	90.7/-19.2	91.9/-2.5	82.7/-4.2	90.7/-6.9
Int.Sim.NEdit	(g,t=0.7)	93.6/-1.6	90.6/-14.6	94.0/-7.1	90.5/-21.8	91.8/-3.8	82.1/-7.8	90.4/-10.3
Int.Sim.	(t=0.4)	**94.1/+6.3**	92.4/+7.3	94.4/0.0	**92.4/+2.6**	92.0/-1.3	83.3/-0.6	91.4/+1.1
Int.Sim.	(g,t=0.4)	**94.1/+6.3**	**93.2/+17.1**	94.4/0.0	91.9/-3.8	**92.1/0.0**	83.3/-0.6	**91.5/+2.3**
Int.Sim.	(r,t=0.4)	93.5/-3.2	92.6/+9.8	94.5/+1.8	91.3/-11.5	91.9/-2.5	**84.0/+3.6**	91.3/0.0

ing. As for using variable cost edit distances we somewhat surprisingly find it to hamper performance, a possible research direction forward would be to induce a more complex edit distance[25] tailored for each semantic category as has been proposed.

Our system, additional results and related resources are freely available for research purposes at: `https://github.com/ninjin/simsem`

Acknowledgments

This work was supported by the Swedish Royal Academy of Sciences and by Grant-in-Aid for Specially Promoted Research (MEXT, Japan). The UK National Centre for Text Mining is funded by the UK Joint Information Systems Committee (JISC).

References

1) Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., Harris, M., Hill, D., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J., Richardson, J., Ringwald, M., Rubin, G. and Sherlock, G.: Gene ontology: tool for the unification of biology, *Nature genetics*, Vol.25, pp. 25–29 (2000).
2) Björne, J., Ginter, F., Pyysalo, S., Tsujii, J. and Salakoski, T.: Scaling up biomedical event extraction to the entire PubMed, *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pp.28–36 (2010).
3) Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology, *Nucleic Acids Research*, Vol.32, pp.D267–D270 (2004).
4) Chowdhury, M. and Lavelli, A.: Disease Mention Recognition with Specific Features, *Proceedings of ACL 2010*, pp.83–90 (2010).
5) Fan, R., Chang, K., Hsieh, C., Wang, X. and Lin, C.: LIBLINEAR: A Library for Large Linear Classification, *Journal of Machine Learning Research*, Vol.9, pp. 1871–1874 (2008).
6) Gerner, M., Nenadic, G. and Bergman, C.: LINNAEUS: A species name identification system for biomedical literature, *BMC Bioinformatics*, Vol.11, No.1, p.85 (2010).
7) Hettne, K., Stierum, R., Schuemie, M., Hendriksen, P., Schijvenaars, B., Mulligen, E., Kleinjans, J. and Kors, J.: A dictionary to identify small molecules and drugs in free text, *Bioinformatics*, Vol.25, No.22, p.2983 (2009).
8) Kim, J., Ohta, T., Tsuruoka, Y., Tateisi, Y. and Collier, N.: Introduction to the bio-entity recognition task at JNLPBA, *Proceedings of JNLPBA 2004*, pp.70–75 (2004).
9) Kim, J., Pyysalo, S., Ohta, T., Bossy, R., Nguyen, N. and Tsujii, J.: Overview of BioNLP Shared Task 2011, *Proceedings of BioNLP Shared Task 2011 Workshop*, pp.1–6 (2011).
10) Kim, J., Wang, Y., Takagi, T. and Yonezawa, A.: Overview of Genia Event Task in BioNLP Shared Task 2011, *Proceedings of BioNLP Shared Task 2011 Workshop* (2011).
11) Maglott, D., Ostell, J., Pruitt, K. and Tatusova, T.: Entrez Gene: gene-centered information at NCBI, *Nucleic Acids Research*, Vol.33, No.suppl 1, p.D54 (2005).
12) Ohta, T., Pyysalo, S. and Tsujii, J.: Overview of the Epigenetics and Post-translational Modifications (EPI) task of BioNLP Shared Task 2011, *Proceedings of BioNLP Shared Task 2011 Workshop* (2011).
13) Ohta, T., Tateisi, Y., Mima, H. and Tsujii, J.: GENIA corpus: an annotated research abstract corpus in molecular biology domain, *Proceedings of HLT 2002*, pp. 73–77 (2002).
14) Okazaki, N. and Tsujii, J.: Simple and Efficient Algorithm for Approximate Dictionary Matching, *Proceedings of Coling 2010*, pp.851–859 (2010).
15) Pyysalo, S., Ohta, T., Rak, R., Sullivan, D., Mao, C., Wang, C., Sobral, B., Tsujii, J. and Ananiadou, S.: Overview of the Infectious Diseases (ID) task of BioNLP Shared Task 2011, *Proceedings of BioNLP Shared Task 2011 Workshop* (2011).
16) Ratinov, L. and Roth, D.: Design challenges and misconceptions in named entity recognition, *Proceedings of CoNLL 2009*, pp.147–155 (2009).
17) Rebholz-Schuhmann, D., Yepes, A., VanMulligen, E., Kang, N., Kors, J., Milward, D., Corbett, P., Buyko, E., Beisswanger, E. and Hahn, U.: CALBC silver standard corpus, *Journal of bioinformatics and computational biology*, Vol.8, No.1, pp.163–179 (2010).
18) Resnik, P.: Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language, *Journal Of Artificial Intelligence Research*, Vol.11, pp.95–130 (1999).
19) Shi, L. and Campagne, F.: Building a protein name dictionary from full text: a machine learning term extraction approach, *BMC bioinformatics*, Vol.6, No.1, p.88 (2005).
20) Stenetorp, P., Pyysalo, S. and Tsujii, J.: SimSem: Fast Approximate String Matching in Relation to Semantic Category Disambiguation, *Proceedings of BioNLP 2011 Workshop*, pp.136–145 (2011).
21) Stenetorp, P., Topić, G., Pyysalo, S., Ohta, T., Kim, J. and Tsujii, J.: BioNLP Shared Task 2011: Supporting Resources, *Proceedings of BioNLP Shared Task 2011 Workshop*, pp.112–120 (2011).
22) Stoyanov, V., Cardie, C., Gilbert, N., Riloff, E., Buttler, D. and Hysom, D.: Coreference resolution with reconcile, *Proceedings of ACL 2010 Short Papers*, pp.156–161 (2010).
23) Thompson, P., Iqbal, S., McNaught, J. and Ananiadou, S.: Construction of an annotated corpus to support biomedical information extraction, *BMC bioinformatics*,

Vol.10, No.1, p.349 (2009).
24) Torii, M., Hu, Z., Wu, C. and Liu, H.: BioTagger-GM: a gene/protein name recognition system, *Journal of the American Medical Informatics Association*, Vol.16, No.2, p.247 (2009).
25) Tsuruoka, Y., McNaught, J. and Ananiadou, S.: Normalizing biomedical terms by minimizing ambiguity and variability, *BMC Bioinformatics*, Vol.9, No.S-3, p.S2 (2008).
26) Tsuruoka, Y. and Tsujii, J.: Improving the Performance of Dictionary-based Approaches in Protein Name Recognition, *Journal of Biomedical Informatics*, Vol.37, pp.461–470 (2004).
27) Wu, C., Yeh, L., Huang, H., Arminski, L., Castro-Alvear, J., Chen, Y., Hu, Z., Kourtesis, P., Ledley, R., Suzek, B., Vinayaka, C., Zhang, J. and Barker, W.: The Protein Information Resource, *Nucleic Acids Research*, Vol.31, No.1, p.345 (2003).
28) Wu, J., Lin, X. and Xie, H.: Porphyromonas gingivalis short fimbriae are regulated by a FimS/FimR two-component system, *FEMS microbiology letters*, Vol.271, No.2, pp.214–221 (2007).