HYENA: Hierarchical Type Classification for Entity Names

Mohamed Amir Yosef[1] *Sandro Bauer*[2] *Johannes Hoffart*[1]
Marc Spaniol[1] *Gerhard Weikum*[1]
(1) Max-Planck-Institut für Informatik, Saarbrücken, Germany
(2) Computer Laboratory, University of Cambridge, UK
{mamir|jhoffart|mspaniol|weikum}@mpi-inf.mpg.de, sandro.bauer@cl.cam.ac.uk

ABSTRACT
Inferring lexical type labels for entity mentions in texts is an important asset for NLP tasks like semantic role labeling and named entity disambiguation (NED). Prior work has focused on flat and relatively small type systems where most entities belong to exactly one type. This paper addresses very fine-grained types organized in a hierarchical taxonomy, with several hundreds of types at different levels. We present HYENA for multi-label hierarchical classification. HYENA exploits gazetteer features and accounts for the joint evidence for types at different levels. Experiments and an extrinsic study on NED demonstrate the practical viability of HYENA.

KEYWORDS: Fine-grained entity types, multi-labeling, hierarchical classification, meta-classification.

1 Introduction

Motivation: Web contents such as news, blogs, etc. are full of named entities. Recognizing them and disambiguating them has been intensively studied (see, e.g., (Finkel et al., 2005; Cucerzan, 2007; Milne and Witten, 2008; Hoffart et al., 2011; Ratinov et al., 2011)). Each entity belongs to one or more *lexical types* associated with it. For instance, an entity such as *Bob Dylan* should be assigned labels of type `Singer`, `Musician`, `Poet`, etc., and also the corresponding supertype(s) (hypernyms) in a type hierarchy, in this case `Person`. Such fine-grained typing of entities can be a great asset for various NLP tasks, e.g. semantic role labeling. Most notably, named entity disambiguation (NED) can be boosted by knowing or inferring a mention's lexical types. For example, noun phrases such as "songwriter Dylan", "Google founder Page", or "rock legend Page" can be easily mapped to the entities Bob Dylan, Larry Page, and Jimmy Page if their respective types `Singer`, `BusinessPerson`, and `Guitarist` are available.

Problem Statement: State-of-the-art tools for named entity recognition like the Stanford NER Tagger (Finkel et al., 2005) compute such lexical tags only for a small set of coarse-grained types: `Person`, `Location`, and `Organization` (plus tags for non-entity phrases of type time, money, percent, and date). There is little literature on *fine-grained typing* of entity mentions (Fleischman and Hovy, 2002; Ekbal et al., 2010; Rahman and Ng, 2010; Ling and Weld, 2012), and these approaches are pretty much limited to flat sets of several dozens of types. Because of the relatively small number of types, an entity or mention is typically mapped to one type only. The goal that we address in this paper is to extend such methods by automatically computing lexical types for entity mentions, using a large set of types from a hierarchical taxonomy with multiple levels. In this setting, many entities naturally belong to multiple types. So we face a *hierarchical multi-label classification* problem (Tsoumakas et al., 2012).

Contribution: This paper introduces *HYENA* (Hierarchical tYpe classification for Entity NAmes). HYENA is a multi-label classifier for entity types based on hierarchical taxonomies derived from WordNet (Fellbaum, 1998) or knowledge bases like YAGO (Suchanek et al., 2007) or Freebase (Bollacker et al., 2008). HYENA's salient contributions are the following:

- the first method for entity-mention type classification that can handle multi-level type hierarchies with hundreds of types and multiple labels per mention;
- extensions to consider cross-evidence and constraints between different types, by developing a meta-classifier demonstrating the superiority of HYENA;
- experiments against state-of-the-art baselines, demonstrating the superiority of HYENA;
- an extrinsic study on boosting NED by harnessing type information.

2 Type Hierarchy and Feature Set

2.1 Fine-grained Type Hierarchy

We have systematically derived a very fine-grained type taxonomy from the YAGO knowledge base (Suchanek et al., 2007; Hoffart et al., 2012) which comes with a highly accurate mapping of Wikipedia categories to WordNet synsets. We start with five broad classes namely `PERSON`, `LOCATION`, `ORGANIZATION`, `EVENT` and `ARTIFACT`. Under each of these superclasses, we pick 100 prominent subclasses. The selection of subclasses is based on the population of the classes: we rank them in descending order of the number of YAGO entities that belong to a class, and pick the top 100 for each of the top-level superclasses. This results in a very fine-grained reference taxonomy of 505 types, organized into a directed acyclic graph with 9 levels in its deepest parts. For instance, this includes fine-grained classifications of an `Adminstrative`

District in order to distinguish between Municipality, Township, Commune, etc. or differentiations of Publications into Books, Periodicals and Magazines.

We are not aware of any similarly rich type hierarchies used in prior work on NER and entity typing. Our approach can easily plug in alternative type taxonomies (e.g. derived from Freebase or DBpedia as in (Ling and Weld, 2012), or from hand-crafted resources such as WordNet).

2.2 Feature Set

For a general approach and for applicability to arbitrary texts, we use only features that are automatically extracted from input texts. We do not use any features that require manual annotations, such as sense-tagging of general words and phrases in training documents. This discriminates our method from some of the prior work which used WordNet senses as features (e.g., (Rahman and Ng, 2010)).

Mention String: We derive the mention string itself (a noun phrase of one or more consecutive words) as well as unigrams, bigrams, and trigrams that overlap with the mention string.

Sentence Surrounding Mention: We derive from a bounded window (size 3) around the mention: all unigrams, bigrams, and trigrams in the sentence along with their distance to the mention, and all unigrams along with their absolute distance to the mention.

Mention Paragraph: We consider the mention paragraph in order to obtain additional topical cues about the mention type. We extract unigrams, bigrams, and trigrams in a bounded window (2000 characters) around the mention (truncated at the paragraph boundaries).

Grammatical Features: We use part-of-speech tags (with/without distance) of the tokens within a bounded window. Further, we resolve the first "he" or "she" pronoun in the same and in the subsequent sentence (including distance) and the closest preceding verb-preposition pair.

Gazetteer Features: We build type-specific gazetteers of words occurring in entity names derived from the YAGO knowledge base. YAGO has a huge dictionary of name-entity pairs extracted from Wikipedia. We automatically construct a binary feature whether the mention contains a word in this type's gazetteer or not. This does not mean determining the mention type(s) (e.g. "Alice" occurs in person subclasses but also in locations, songs, organizations, etc.).

3 Classifier

3.1 Hierarchical Classifier

Based on the feature set defined in the previous section, we build a set of type-specific classifiers using the SVM software liblinear (Fan et al., 2008; Chang and Lin, 2011). As our YAGO-based type system integrates WordNet and Wikipedia categories, we obtain ample training data from Wikipedia effortlessly, by following Wikipedia anchor texts to the corresponding YAGO entities.

For each type, we consider Wikipedia mentions (and their context, cf. Section 2.2) of the type's instances as positive training samples. For discriminative learning, we use all siblings in the type hierarchy as negative samples. As the subclasses of type t do not necessarily cover all entities, we add a subclass Others to each non-leaf type. Positive samples for Others are instances of type t that do not belong to any of its subclasses. Conversely, the classifiers for non-leaf nodes include all instances of their subtypes as positive samples (with full weight). HYENA performs type-specific classification in a top-down manner. A mention is assigned to all types for which the classifier signals acceptance. If rejected, classification is stopped at this level.

3.2 Meta Classifier

HYENA uses a global threshold θ for accepting to a class. Using a single parameter for all types is not fully satisfying, as different types may exhibit very different characteristics. So the optimal acceptance threshold may be highly type-dependent. To overcome this limitation, we devised a meta classifier that ranks the types for each test mention by decreasing confidence values and then predicts the "right" number of top-n labels to be assigned to a mention, similar to the methodology of (Tang et al., 2009). We use the confidence values of the type-specific classifier ensemble as meta-features, and train a multi-class logistic regression classifier to obtain a suitable value n of features. We combine the base classifiers and the meta classifier by first running the entire ensemble top-down along the type hierarchy, and then letting the meta model decide on how many of the highest-scoring types we accept for a mention.

4 Experiments

4.1 Setup

System: The described methods are implemented in HYENA. The Stanford NLP tools are used to identify mentions of named entities and to extract grammatical features from the context.
Data: We used the English Wikipedia edition as of 2012-05-02. In order to obtain ground-truth type labels, we exploited the links to other Wikipedia articles, resolved the corresponding YAGO2 entity and retrieved the semantic types. For example, from the Wikipedia markup:

> "In June 1989, Obama met [[Michelle Obama|Michelle Robinson]] when he was employed as a summer associate at the Chicago law firm of [[Sidley Austin]]"

the following YAGO2 entities are assigned:

> Michelle Robinson → http://yago-knowledge.org/resource/Michelle_Obama
> Sidley Austin → http://yago-knowledge.org/resource/Sidley_Austin

HYENA is trained on 50,000 randomly Wikipedia articles selected, containing around 1.6 million entity mentions. 92% of the corresponding entities belong to at least one of our 5 top-level types, with 11% belonging to at least two top-level types. Testing of HYENA is performed on 10,000 randomly selected Wikipedia articles withheld from the same Wikipedia edition and disjoint from the training data. All experimental data is available at http://www.mpi-inf.mpg.de/yago-naga/hyena/.
Performance Measures: We report micro- and macro-evaluation numbers for precision, recall and F1 scores. Let T be the set of all types in our hierarchy, and let I_t be the set of instances tagged with type t, and let $\hat{I}_t$ the set of instances that are predicted to be of type t. The measures used are:

$$Precision_{micro} = \frac{\sum_{t \in T} \left| I_t \cap \hat{I}_t \right|}{\sum_{t \in T} \left| \hat{I}_t \right|} \quad and \quad Recall_{micro} = \frac{\sum_{t \in T} \left| I_t \cap \hat{I}_t \right|}{\sum_{t \in T} \left| I_t \right|}$$

$$Precision_{macro} = \frac{1}{|T|} \sum_{t \in T} \frac{\left| I_t \cap \hat{I}_t \right|}{\left| \hat{I}_t \right|} \quad and \quad Recall_{macro} = \frac{1}{|T|} \sum_{t \in T} \frac{\left| I_t \cap \hat{I}_t \right|}{\left| I_t \right|}$$

Competitors: We identified the methods by (Fleischman and Hovy, 2002) referred to as *HOVY*, (Rahman and Ng, 2010) referred to as *NG*, and *FIGER* by (Ling and Weld, 2012) for comparison (cf. Section 6). We conducted experiments on the competitors' datasets to avoid re-implementation and to give them the benefit of their original optimization and tuning.

	Macro			Micro		
	Prec.	Rec.	F1	Prec.	Rec.	F1
5 Top-level Types	0.941	0.922	0.932	0.949	0.936	0.943
All 505 Types	0.878	0.863	0.87	0.913	0.932	0.922

Table 1: Overall Experimental Results for HYENA on Wikipedia 10000 articles

		Macro			Micro		
		Prec.	Rec.	F1	Prec.	Rec.	F1
5 Top-level Types	*HOVY*	0.522	0.464	0.491	0.568	0.51	0.537
	HYENA	**0.941**	**0.922**	**0.932**	**0.949**	**0.936**	**0.943**
All 505 Types	*HOVY*	0.253	0.18	0.21	0.405	0.355	0.378
	HYENA	**0.878**	**0.863**	**0.87**	**0.913**	**0.932**	**0.922**

Table 2: Results of HYENA vs *HOVY* (trained and tested on Wikipedia 10000 articles)

4.2 Multi-label Classification

We present multi-label experiments that are geared for high precision and high recall. Experiments are performed against ground truth coming from Wikipedia, the BBN Pronoun Coreference Corpus and Entity Type Corpus (LDC2005T33)and the FIGER-Gold dataset.

4.2.1 HYENA experiments on Wikipedia

The results of our HYENA approach on Wikipedia are shown in Table 1. HYENA achieves very high F1 scores of around 94% for its 5 top-level types. Evaluated against the entire hierarchy, F1 scores are still remarkably high with F1 scores of 87% and 92% for macro and micro evaluations, respectively. The slightly weaker results for the macro evaluation are explainable by our fine-grained hierarchy, which also contains a few "long-tail" types.

In order to compare against *HOVY*, we emulated their method within the HYENA framework. This is done by specifically configuring the feature set, and using the same training and testing instances as for HYENA. Results are shown in Table 2. HYENA significantly outperforms *HOVY*. Similar to the results reported in (Fleischman and Hovy, 2002) *HOVY* shows decent performance for the 5 top-level types, but performance sharply drops for subtypes at deeper levels.

4.2.2 HYENA Experiments on FIGER-GOLD

The FIGER-GOLD dataset consists of 18 news reports from a university website, as well as local newspapers and specialized magazines (Ling and Weld, 2012). The test dataset was annotated with at least one label per mention. This resulted in a total of 434 sentences with 563 entities having 771 labels coming from 42 out of the 112 types. The original evaluation for FIGER was instance-based. In order to compare against HYENA, a per-type evaluation is needed. To this end, we created a per-type based classification of FIGER based on their output data. Since the distribution of mentions on different types in the FIGER dataset is heavily skewed (e.g. 217 of the 562 entities are of type `PERSON` without finer-grained subtype annotation) we cover in our evaluation the most 10% populated classes (covering around 70% of the tags). These classes were then mapped onto the hierarchy of HYENA. Since all instances in the FIGER-GOLD dataset are tagged with at least one class, we ran HYENA in two configurations: without any modification as before (using a classifier trained to deal with abstract concepts, e.g. `Chinese Philosophy`, that are of generic type `ENTITY_OTHER`) as well as by enforcing the assignment

	Macro			Micro		
	Prec.	Rec.	F1	Prec.	Rec.	F1
FIGER	**0.75**	0.743	0.743	**0.828**	**0.838**	**0.833**
HYENA	0.745	0.631	0.684	0.815	0.645	0.72
HYENA (at least one tag)	0.724	**0.801**	**0.75**	0.788	0.814	0.801

Table 3: Results of HYENA vs *FIGER* (trained on Wikipedia and tested on FIGER-Gold)

	Macro			Micro		
	Prec.	Rec.	F1	Prec.	Rec.	F1
NG (trained on BBN)	0.859	0.864	0.862	0.812	0.871	0.84
HYENA (trained on Wikipedia)	**0.943**	0.406	0.568	**0.932**	0.371	0.531
HYENA (trained on Wikipedia, at least one tag)	0.818	0.671	0.737	0.835	0.632	0.719
HYENA (trained on BBN)	0.916	**0.909**	**0.911**	0.919	**0.881**	**0.899**

Table 4: Results of HYENA vs *NG* (tested on BBN Corpus)

of at least one class for all instances (referred to as "at least one tag").

Results are shown in Table 3. In the standard configuration, HYENA shows precision scores close to *FIGER*. However, HYENA suffers from the training against abstract concepts. In the second configuration, both systems achieve results in the same range with slight advantages for *FIGER* on micro-average and overall better results of HYENA on macro-average. However, 771 type labels for 562 entity mentions (not entities) is only a very moderate amount of multi-label classification. This is disadvantageous for HYENA, which has been designed for data where the number of labels per mention is higher.

4.2.3 HYENA Experiments on BBN

The BBN Pronoun Coreference and Entity Type Corpus consists of 2311 manually annotated documents. Since *NG* exploits WordNet word-senses for disambiguation, the corpus is restricted to those 200 documents (160 training, 40 testing) that have corresponding annotations. For comparison against *NG* we performed a mapping onto the hierarchy of HYENA. Among the 16 types for the *NG* dataset (cf. (Rahman and Ng, 2010)), there are 8 non-entity types (e.g. `Date`) and 5 descriptor types (`_DESC`) which cannot be mapped. This resulted in mapping the 3 top-level types: `Person`, `Organization` and `GPE` (country, city, states, etc.). Similar to the FIGER-GOLD dataset, there are no unclassified mentions in the BBN corpus. Hence we ran HYENA in three configurations: standard ("trained on Wikipedia"), enforcing at least one type label to be assigned ("trained on Wikipedia, at least one tag") and HYENA trained on the *NG* training set ("trained on BBN").

Results on the BBN dataset exhibit high precision of HYENA already with its standard configuration (cf. Table 4). However, it suffers from low recall in this setting, due to training against abstract concepts. When enforcing HYENA to assign at least one tag, F1 scores strongly improve. In the third configuration, the fairest side-by-side comparison, we clearly outperform *NG*.

4.3 Meta-Classification

In use-cases for type labeling (e.g. NED), precision is often more important than recall. This is particularly demanding for types that suffer from data sparsity (less prominent and/or less populated types) deep in the type hierarchy. For example in NED, it may be crucial to distinguish

		Macro			Micro		
		Prec.	Rec.	F1	Prec.	Rec.	F1
All 505 Types	HYENA	0.878	**0.863**	**0.87**	0.913	**0.932**	**0.922**
	HYENA + meta-classifier	**0.89**	0.837	0.862	**0.916**	0.914	0.915

Table 5: Performance gain in precision by meta-classification

	Macro			Micro		
	Prec.	Rec.	F1	Prec.	Rec.	F1
HYENA	0.673	**0.638**	**0.644**	0.659	**0.681**	**0.67**
HYENA + meta-classifier	**0.693**	0.619	0.638	**0.674**	0.66	0.667

Table 6: Meta-classifier impact on the 5% worst-performing classes

a `Painter` from a `Musician`. When applied, meta classification (see Section 3.2 for details) improves macro-precision over all 505 types by more than 1% (cf. Table 5). When focusing on the 5% types that performed worst without it, we even gained more than 2% in precision, as shown in Table 6. The top-5 winners in this group gain from 5% up to 13%.

4.4 HYENA Feature Analysis

In addition to a comprehensive feature set, HYENA exploits a large amount of training data and the gazetteer features derived from YAGO. To assess the impact of each asset, we varied the number of training instances and en-/disabled gazetteer features (cf. Table 7). Precision and recall improve from a larger training corpus, particularly for sparsely populated types. When gazetteer features are disabled, performance drops significantly.

5 Extrinsic Study on Named Entity Disambiguation

We conducted an extrinsic study on harnessing HYENA for NED, based on a state-of-the-art NED tool, AIDA by (Hoffart et al., 2011). This NED method uses a combination of contextual similarity and entity-entity coherence for disambiguation. In order to speed up its computationally expensive graph algorithms, it is desirable to prune the search space. Hence, we use the type predictions by HYENA for pruning (e.g. for the sentence "He was born in Victoria" and the mention "Victoria", the entities of type `Person`, `River` and `Lake` should be dropped). To this end, we use the confidence scores of HYENA to remove entities of types with type scores below some threshold θ. Our technique proceeds in three steps:

1. Invoke HYENA on the mention to obtain the predicted types and confidence scores.
2. Generate entity candidates using AIDA and its underlying name-entity dictionary.
3. For each candidate, if there is no overlap between the entity types and the predicted mention types with confidence greater than or equal to θ, drop the candidate.
4. Run AIDA on the reduced candidate space.

When dropping the correct entity, a mention becomes *unsolvable*. We vary the relaxation parameter θ to investigate search space rdecution versus mentions that are rendered *unsolvable*. We performed our experiment on the extended CoNLL 2003 NER dataset with manual entity annotations from (Hoffart et al., 2011). With a pruning threshold of $\theta = -1$, we can prune almost 40% of all entities while rendering less than 8% of the mentions unsolvable (cf. Table 8). The search space reduction of 40% actually results in a much larger saving in run-time because the graph algorithm that AIDA uses for NED has super-linear complexity (NP-hard in the worst case, but typically $O(n \log n)$ or $O(n^2)$ with appropriate approximation algorithms).

Size of training set (# of articles)	5 Top-level Types			All 505 Types		
	Prec.	Rec.	F1	Prec.	Rec.	F1
50,000	0.949	0.936	0.942	0.913	0.932	0.922
20,000	0.937	0.924	0.93	0.893	0.917	0.905
5,000	0.92	0.903	0.912	0.869	0.89	0.879
50,000 (without gazetteers)	0.915	0.825	0.868	0.82	0.718	0.766

Table 7: Micro-average impact of varying the number of Wikipedia articles used for training

Threshold	% dropped Entities	% unsolvable Mentions	avg. Document Prec.	avg. Mention Prec.
0.0	49.2	16.1	0.659	0.639
−0.5	45.7	12.3	0.738	0.713
−1.5	28.8	4.7	0.791	0.779
−2.5	17.7	2.2	0.802	0.798
AIDA	0	0	0.82	0.823

Table 8: Impact of Varying Type Prediction Confidence Threshold on NED Results

6 Related Work

There is little prior work on the task of classifying named entities, given in the form of (still ambiguous) noun phrases, onto fine-grained lexical types. (Fleischman and Hovy, 2002) has been the first work to address type granularities that are finer than the handful of tags used in classical NER work (person, organization, location, date, money, other – see, e.g., (Wacholder et al., 1997; Alfonseca and Manandhar, 2002; Cunningham, 2002; Finkel et al., 2005)). It considered 8 sub-classes of the `Person` class, and developed a decision-tree classifier. (Ekbal et al., 2010) developed a maximum entropy classifier using word-level features from the mention contexts, but experimental results are flagged as non-reproducible in the ACL Anthology. (Rahman and Ng, 2010) considered a two-level type hierarchy consisting of 29 top-level classes and a total of 92 sub-classes. These include many non-entity types such as date, time, percent, money, quantity, ordinal, cardinal, etc. The method uses a rich set of features, including WordNet senses of noun-phrase head words in mention contexts. (Giuliano, 2009) proposed an SVD-based latent topic model with a semantic kernel that captures word proximities. The method was applied to a set of 21 different types; each mention is assigned to exactly one type. The work of (Ling and Weld, 2012) considered a two-level taxonomy with 112 tags taken from the Freebase knowledge base, forming a two-level hierarchy with top-level topics and 112 types (with entity instances). (Ling and Weld, 2012) trained a CRF for the joint task of recognizing entity mentions and inferring type tags. The feature set included the ones used in earlier work (see above) plus patterns from ReVerb (Fader et al., 2011).

7 Conclusions

We presented HYENA for fine-grained type classification of entity mentions. In contrast to prior methods, we can deal with hundreds of types in a multi-level hierarchy, and consider that a mention can have many different types. In experiments, HYENA outperformed state-of-the-art competitors even on their original datasets and improved efficiency of NED by reducing the search space.

Acknowledgements

This work is supported by the 7^{th} Framework IST programme of the European Union through the focused research project (STREP) on Longitudinal Analytics of Web Archive data (LAWA) – contract no. 258105.

References

Alfonseca, E. and Manandharm, S. (2002). An Unsupervised Method for General Named Entity Recognition and Automated Concept Discovery. In *Proc. of the 1st International Conference on General WordNet*.

Bollacker, K. D., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. of SIGMOD Conference*, pages 1247–1250.

Bunescu, R. and Pasca, M. (2006). Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proc. of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL '06)*, pages 9–16.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on wikipedia data. In *Proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (EMNLP-CoNLL), pages 708–716.

Cunningham, H. (2002). Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254.

Ekbal, A., Sourjikova, E., Frank, A., and Ponzetto, S. P. (2010). Assessing the challenge of fine-grained named entity recognition and classification. In *Proc. of the 2010 Named Entities Workshop*, (NEWS '10), in conjunction with *the 48th Annual meeting of the Association for Computational Linguistics*. (ACL '10), pages 93–101, Stroudsburg, PA, USA.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proc. of the Conference on Empirical Methods in Natural Language Processing* (EMNLP '11), pages 1535–1545.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics*, (ACL '05), pages 363–370, Stroudsburg, PA, USA.

Fleischman, M. and Hovy, E. (2002). Fine grained classification of named entities. In *Proc. of the 19th International Conference on Computational Linguistics - Volume 1*, (COLING '02), pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Giuliano, C. (2009). Fine-Grained Classification of Named Entities Exploiting Latent Semantic Kernels. In *Proc. of the 13th Conference on Computational Natural Language Learning*, (CoNLL '09) pages 201–209.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proc. of the Conference on Empirical Methods in Natural Language Processing* (EMNLP '11), pages 782–792, Edinburgh, Scotland, United Kingdom 2011.

Hoffart, J., Suchanek, F., Berberich, K. and Weikum, G. (2012). YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Special issue of the Artificial Intelligence Journal* 2012.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proc. of the 5th annual International Conference on Systems Documentation*, (SIGDOC '86), pages 24–26, New York, NY, USA. ACM.

Ling, X. and Weld, D. S. (2012). Fine-grained entity recognition. In *Proc. of the 26th AAAI Conference on Artificial Intelligence*, Toronto, Ontario, Canada.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.

Milne, D. and Witten, I. H. (2008). Learning to link with wikipedia. In *Proc. of the 17th ACM conference on Information and knowledge management*, (CIKM '08), pages 509–518, New York, NY, USA. ACM.

Rahman, A. and Ng, V. (2010). Inducing fine-grained semantic classes via hierarchical and collective classification. In *Proc. of the 23rd International Conference on Computational Linguistics* (COLING 2010), pages 931–939.

Ratinov, L.-A., Roth, D., Downey, D., and Anderson, M. (2011). Local and global algorithms for disambiguation to wikipedia. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics* (ACL '11), pages 1375–1384.

Suchanek, F., Kasneci, G., and Weikum, G. (2007). YAGO: A core of semantic knowledge - unifying WordNet and Wikipedia. In Williamson, C. L., Zurko, M. E., and Patel-Schneider, Peter F. Shenoy, P. J., editors, In *Proc. of the 16th International World Wide Web Conference* (WWW '07), pages 697–706, Banff, Canada. ACM.

Tang, L., Rajan, S., and Narayanan, V. K. (2009). Large scale multi-label classification via metalabeler. In *Proc. of the 18th International World Wide Web Conference*, (WWW '09), pages 211–220, New York, NY, USA. ACM.

Tsoumakas, G., Zhang, M.-L., and Zhou, Z.-H. (2012). Introduction to the special issue on learning from multi-label data. *Machine Learning*, 88(1-2):1–4.

Wacholder, N., Ravin, Y., and Choi, M. (1997): Disambiguation of Proper Names in Text. In *Proc. of the 5th Conference on Applied Natural Language Processing* (ANLP '97), pages 202-208.