Multilingual Fine-Grained Entity Typing

van Erp, M.G.J.; Vossen, P.T.J.M.

published in
Language, Data, and Knowledge
2017

DOI (link to publisher)
10.1007/978-3-319-59888-8_23

document version
Publisher's PDF, also known as Version of record

document license
Article 25fa Dutch Copyright Act

Multilingual Fine-Grained Entity Typing

Marieke van Erp(✉) and Piek Vossen

Computational Lexicology and Terminology Lab, The Network Institute,
Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
{marieke.van.erp,piek.vossen}@vu.nl

Abstract. Many entity recognition approaches classify recognised entities into a limited set of coarse-grained entity types. However, for deeper natural language analysis and end-user tasks, fine-grained entity types are more useful. For example, while standard named entity recognition may determine that an entity is a person knowing whether that entity is a politician or an actor is important for determining whether, in a subsequent relation extraction task, a relation should be acts or governs. Currently, fine-grained entity typing has only been investigated for English. In this paper, we present a fine-grained entity typing system for Dutch and Spanish using training data extracted from Wikipedia and DBpedia. Our system achieves comparable performance to English with an F_1 measure of .90 on over 40 types for both Dutch and Spanish.

1 Introduction

Entity typing is the task of assigning types (also called classes) to previously recognised entity mentions in text. Traditionally, a limited set of types is employed (e.g. Person, Location, Organisation, Miscellaneous) [12] but for many NLP tasks more fine-grained types have been proven useful [16,23]. Fine-grained entity typing can support, coreference resolution, relation extraction, entity linking (e.g. distinguishing between Douglas Adams the author or the American Football Player) and dark entity classification (i.e. determining the class of those entities not described in a knowledge base). For English, the task of fine-grained entity typing has received some attention (cf. [3,5,9,24]), but to the best of our knowledge, other languages such as Spanish and Dutch have thus far not been addressed.

One of the main issues in fine-grained entity typing is noise in the training data, as training samples are usually generated automatically [5] by assigning Freebase types to entity mentions extracted from news or Wikipedia. [17] seek a solution to this problem by proposing a cleaning method. We mitigate this issue by choosing a more restrictive, but cleaner type hierarchy for generating our training samples, namely DBpedia [1].

In this paper, we present a distantly supervised approach to fine-grained entity typing for Dutch and Spanish based on Wikipedia and DBpedia. Our approach is inspired by [9] (further explained in Sect. 3) who also generate training data from Wikipedia, but instead of using Freebase types, we use the DBpedia

J. Gracia et al. (Eds.): LDK 2017, LNAI 10318, pp. 262–275, 2017.
DOI: 10.1007/978-3-319-59888-8_23

type hierarchy. There are three advantages to this: (1) as DBpedia is derived from Wikipedia, there is a direct link between the two sources leading directly to the entity type, and (2) DBpedia only assigns one type to an entity, thus leading to less noise in the training data, and (3) DBpedia contains language and cultural specific data that can be leveraged for training and testing. Furthermore, we are the first to employ the Fasttext algorithm [2,8] to the task of entity typing and to apply it to morphologically richer languages than English.

To compare our approach to previous work, we report results on a subset of DBpedia's 685 types mapped to the types reported on in [5]. As well as on the full DBpedia class hierarchy. Our code and experimental setup are available at https://github.com/cltl/multilingual-finegrained-entity-typing.

The contributions of this work are threefold:

1. state-of-the-art fine-grained entity typing models for Dutch and Spanish;
2. an extensible system for entity typing for other languages using Wikipedia and DBpedia; and
3. an analysis of the DBpedia entity type system for this task.

The remainder of this paper is organised as follows. In Sect. 2, we discuss related work. In Sect. 3, we detail our data preparation and algorithm used. In Sect. 4, our experiments and results are presented, followed by a discussion in Sect. 5. We conclude with conclusions and future work in Sect. 6.

2 Related Work

Entity recognition and classification (NERC) has been a long-standing and popular task in the natural language processing community [12]. However, most work has focused on a limited set of entity types, for example, four in the CoNLL NER campaigns (Person, Location, Organisation and Miscellaneous) [19] and seven main types in the ACE campaigns (Person, Organization, Location, Facility, Weapon, Vehicle and Geo-Political Entity) [10]. Several extensions to these were proposed such as hierarchy of 150 entity types [18] and unsupervised approaches that could infer new entity types from text [4,6,7,13]. The increase in external knowledge bases such as DBpedia and Freebase has allowed research into fine-grained entity typing gained to expand.

In the FIGER system [9], a two-level hierarchy consisting of 112 entity types is proposed. The training data is generated using Wikipedia, where the wikilink anchor text is extracted as an entity mention which map it to its corresponding Freebase entity types, following [15]. For each entity mention, a feature vector is generated that contains information about the entity tokens, the word shape (capitalisation patterns), part-of-speech tags, its length, and words before and after the entity mention. They then train a perceptron algorithm which is evaluated on a manually annotated test set of 18 articles taken from a student newspaper. Their system achieves an F_1 score of .70.

The FIGER hierarchy initially consisted of two levels, but was extended to three levels by [5]. They also use an automated method to generate training data but use news articles instead of Wikipedia. In follow-up work [24], their best system achieves an F_1 of .72 in classifying 86 entity types using an embedding method. The features used are loosely based on [9] and they evaluate on 77 manually annotated news articles from the OntoNotes corpus [20]. In the approach presented in [24], an embedding is learnt for each label and feature which allows for information sharing between labels.

The HYENA system [25] also employs Wikipedia to gather training data. They use a type hierarchy consisting of 505 types, up to 9 levels deep that is induced from YAGO[1] and WordNet.[2] For each type, an SVM classifier is trained, after which the results of the individual classifiers are put through a meta-classifier to define the final top-n types for an entity mention. Their system achieves a macro F_1 score of .93 on the 5 top-level types and .87 on the full 505 types.

Word embeddings and the FIGER hierarchy are also used by the FIGMENT system [21,22]. In this system, the 112 FIGER types are learnt through different character, word and entity level models using a training and test dataset generated from a web crawl. They also analyse their system's performance for dark entities, where they achieve an F_1 of around .51. Related to this is the PEARL system [14] which explicitly tries to detect entity types for emerging entities using integer linear programming (ILP). They use the YAGO2 type system, which is derived from WordNet.

The FINET system [3] generates training data from WordNet. Their system uses various patterns on parsed text to determine types. Their type system consists of more than 16,000 types, which are mapped to two other type system consisting of 505 and 200 types to determine the first three levels in the type hierarchy (coarse-grained types, fine-grained types and super fine-grained types).

In Table 1 we summarise the characteristics of previous approaches to fine-grained entity typing.

Table 1. Characteristics of previous approaches to fine-grained entity typing.

System	Type system	# Types	Depth	Global/Contextual	Approach	Training data
FIGER	Freebase	112	2	Global	Perceptron	Wikipedia
FINET	WordNet	1,000+	9	Contextual	Patterns	WordNet
FIGMENT	Freebase	86	3	Both	Embeddings	Websites
GFT	Freebase	86	3	Contextual	Embeddings	News
HYENA	YAGO	505	9	Contextual	SVM	Wikipedia
PEARL	YAGO2	NA	NA	Contextual	ILP	News
Hovy 2014 [7]	Induced	1,000+	NA	Contextual	HMM	News

[1] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/.

[2] https://wordnet.princeton.edu/.

An important distinction to be made is whether the system is aimed at global entity typing or contextual entity typing. In global entity typing, multiple types can be assigned to an entity. This is generally used to generate entity type information for knowledge bases where the goal is to describe an entity in the most detailed fashion possible. In contextual entity typing, usually fewer types are assigned, and the goal is to decide for a particular sentence what entity type is meant. FIGER and [5] are aimed at global entity typing, FINET at contextual typing and FIGMENT can do both.

However, none of these systems has been applied to languages other than English.

3 System Description

In Fig. 1, a systematic overview is given of our system. As in [9], we generate labelled training instances from Wikipedia, but where they gather the entity types from Freebase, we use the DBpedia type system to obtain associated types for each entity. Then, feature vectors are generated that contain information about the word shape of the entity and its context (a detailed overview of the features is given in Table 2) which is used to train a model. The model is then applied to a new instance, for which it is to predict a type. In the remainder of this section, each step is explained in more detail.

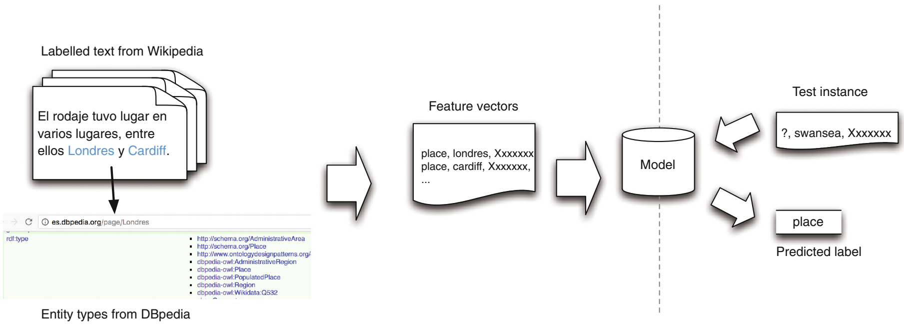

Fig. 1. System overview

3.1 Data Preparation

We use the Dutch and Spanish 2017-02-01 Wikipedia XML dumps.[3] The text was extracted from each dump using the Wikiextractor tool[4] while preserving the wikilinks. For each wikilink, we extract the anchor as our entity mention,

[3] https://dumps.wikimedia.org/backup-index.html.
[4] https://github.com/attardi/wikiextractor.

and we look up the corresponding DBpedia type through a Wikipedia-DBpedia mapping.[5]

There are three main reasons for using the DBpedia type hierarchy: (1) we presume that a single type per entity results in cleaner training data, (2) whilst some other type hierarchies such as yago, umbel and schema.org are well connected to the English DBpedia, they are less commonly used in the Dutch and Spanish DBpedias, limiting their use to generate training examples from such data, (3) as DBpedia is derived from Wikipedia, there is a direct link through which the entity type can be retrieved.

For each entity, we generate a feature vector containing the mention, the head, and some context, following [5]. We leave out the dependency and topic related features due to processing constraints. A sample of the features for both Dutch and Spanish is presented in Table 2.

Table 2. Description of the extracted features.

Feature	Description	Dutch example	Spanish example
Mention	The entity phrase	San Francisco	Benedict Cumberbatch
Head	The syntactic head of the entity phrase	Francisco	Cumberbatch
Non-head	The non-head tokens in the entity phrase	San	Benedict
Entity shape	The word shape of the words in the entity phrase	Aaa Aaaaaaaaa	Aaaaaaaa Aaaaaaaaaaa
Trigrams	Character trigrams in the entity head	_Fr Fra ran anc nci cis isc sco co_	_Cu Cum umb mbe ber erb rba bat atc tch ch_
Word before	The word before the entity phrase	te	actor
Word after	The word after the entity phrase	Californië	fue

To compare our results to those in previous work, we mapped the DBpedia type hierarchy to the entity typing hierarchy used in [5, 24]. Out of the 86 types that were present, 9 types could not be mapped to the DBpedia type hierarchy.[6]

[5] Using the wikilinks and instance types dumps from the latest DBpedia, version 2016-04 http://wiki.dbpedia.org/downloads-2016-04.

[6] The types we could not map were the following: `location/structure/government`, `organization/stock_exchange`, `other/health`, `other/living_thing`, `other/product/car`, `other/product/computer`, `person/education`, `person/education/student`, `person/education/teacher`.

To check the mappings, we compared some of the entity mentions and their types from the GFT dataset [5] to entity mentions and their respective DBpedia types.

As there are currently no fine-grained entity type gold standard datasets available for Dutch and Spanish, we split the generated data into training and test sets by random stratified sampling, i.e. in both the training and test sets the class distribution is proportionally equal. We chose the test sample as 1/3 of the total dataset. In Table 3, we present some statistics of our datasets. On average, each entity mention occurs about 5 times in the Spanish dataset and about 7 times in the Dutch dataset.[7]

Table 3. Dataset statistics

	Dutch		Spanish	
	Instances	Unique mentions	Instances	Unique mentions
Training GFT types	1,011,810	143,793	561,249	104,174
Test GFT types	498,355	93,735	276,437	69,137
Training DBpedia types	2,088,381	256,502	1,066,644	209,653
Test DBpedia types	1,028,607	166,001	525,363	138,482

3.2 Model Construction

We use the fastText [2,8][8] algorithm in our classification experiments. The fastText algorithm is a linear algorithm inspired by the word2vec cbow model [11] that utilises a hierarchical softmax function to speed up computations as it represents more frequent classes in the dataset at a lower depth than more infrequent ones. Representations are learnt for character n-grams, and words are represented as the sum of the n-gram vectors, which helps in covering morphologically rich languages, words that do not occur often and potentially entity mentions that do not occur in the training corpus. As in the algorithm used by [24], fastText can share information between features, which can be particularly useful for classes with few examples.

4 Experiments and Results

We ran two sets of experiments for each language: in the first, we only take the GFT types into account, in the second, we consider all DBpedia types present in our datasets.

[7] Although there is more text in the Spanish DBpedia, we only included a sample here to showcase the adaptability of the approach to other languages.
[8] https://github.com/facebookresearch/fastText.

4.1 GFT Types

There are some differences between our setup and the setup of [5,24]. First, their gold standard dataset is manually labelled and entities can contain multiple labels, which, if they follow [5] are separated into different instances.[9]

We only assign a single type to an entity, and we take a sample from the automatically generated data from Wikipedia for testing. The single type per entity premise does mean that we do not generate global type information for entities (i.e. multiple types per entity which would be useful for knowledge base creation), but we do only focus on the 'main' type of an entity according to our training dataset. How much this overlaps with the contextual types that are generated in [5,24] can only be investigated with a gold standard dataset for Dutch and Spanish. [24] report a micro-averaged F_1 of 72.98 for their best system using 86 fine-grained types.

In Table 4, we present our scores per level of depth in the hierarchy. The results for the coarse-grained entity types (depth 1) are near perfect, reaching scores in the high 90s, but the fine-grained (depth 2) and super fine-grained (depth 3) are also quite high.[10] Not all GFT types are covered in our datasets. We could determine various causes for this. The first is that the type information in the Dutch and Spanish DBpedias seems to be less extensive than for the English DBpedia, and the types file only contains the most specific entity type assigned to a resource, not its supertypes. We will delve deeper into these issues in Sect. 5.

Table 4. Macro-averaged Precision (P), Recall (R) and F_1 scores

Types	Dutch			Spanish		
	P	R	F_1	P	R	F_1
1 (4 types)	.98	.98	.98	.97	.97	.97
2 (33/24 types)	.92	.90	.91	.91	.90	.90
3 (24/20 types)	.89	.91	.90	.87	.90	.88
Overall (59/41 types)	.93	.88	.90	.92	.88	.90
Only dark entities (59/41 types)	.67	.56	.60	.74	.63	.66
All DBpedia types overall (269/143 types)	.68	.52	.57	.83	.75	.78
All DBpedia types only dark entities (266/143 types)	.50	.41	.44	.44	.37	.39

Whilst the test set is a separate set that is held out from the training set, some entity mentions may overlap between the two, for example for popular entities that occur with a high frequency in Wikipedia. To gain an insight into how our

[9] If an entity X has types `location/structure` and `organisation/education` assigned to it, two instances are generated namely X, `location/structure` and X, `organisation/education`.

[10] The number of types from levels 1–3 do not add up to the total number of types as some of the higher level types are not present on their own, such as `other`.

approach performs on unknown, or dark, entities, we removed all entity mentions from the test set that also occurred in the training dataset and evaluated our models on only those. This results in an F_1 score of .60 for Dutch and .66 for Spanish. This does leave major room for improvement, but these results are a bit higher than those reported for dark entities in [21] (F_1.51). Furthermore, it is unlikely that all entities in a task are unknown.

Table 5 in the Appendix presents the results of our system per class for Dutch and Spanish. The results in this table highlights differences in the type distribution between Dutch and Spanish, and ensuing differences in performance (cf. `other/art/stage`). For most types, despite there not being that much training data, the performance is still quite reasonable. The performance on `person/doctor`, for example, is quite reasonable with an F_1 score of .81, but other types seem much less well defined (for example `person/legal` for which more training examples were available). Further analysis of the entity types is needed to determine how this could be improved.

4.2 DBpedia Types

As the training data contains more types than those defined in the GFT hierarchy, we also ran an experiment using those types. In total, the DBpedia ontology contains 685 types, but these were not all present in our data. Upon inspecting the Dutch DBpedia types file, we found that only 274 types are present there. For Spanish, we only find 147 in the DBpedia types file. One reason for this is that the typing file only contains the most specific type that is assigned to an instance, for example for http://nl.dbpedia.org/resource/Old_Amsterdam, the most specific type is `cheese`, but the mapping only goes only to its parent node `Food`. Furthermore, the Dutch and Spanish DBpedias are less likely to report on entities of type `AustralianRulesFootballPlayer` or `NationalCollegiateAthleticAssociationAthlete`.

The last two rows of Table 4 presents the results of the DBpedia types experiments. The first DBpedia results row is on the full test set, the second reports results only on the dark entities. For Spanish, the experiments with the DBpedia types still perform quite well, but for Dutch they drop dramatically compared to the experiments using the GFT types. This can partially be explained by the fact that for Dutch there are twice as many types in the dataset, making the typing problem more complex. The drop is less steep for the dark entities for Dutch though, indicating that the approach somehow captures information about unknown entities better for Dutch than for Spanish.

5 Discussion

Whilst the results, in particular for the smaller GFT hierarchy, are comparable to English, there is room for improvement on typing using more extended hierarchies such as the DBpedia type hierarchy. In this section, we discuss causes for the lower scores using the DBpedia typing and possible solutions.

Dataset Extension

As mentioned in the previous section, the DBpedia entity types file only lists the most specific type for a given entity. The availability of this file allows for quick adaptation to another language without having to download the entire DBpedia version, but it does have limitations in that its coverage is lower. In future work, we aim to investigate including more super- and sub-types in generating the training and test samples to increase the coverage of types.

DBpedia has other type information available besides the DBpedia type hierarchy. We chose to utilise the DBpedia type hierarchy because each DBpedia resource has only one set of types possible subtypes assigned to it. For example, `dbpedia:Arnold_Schwarzenegger`[11] has only `dbo:Agent`[12] `dbo:Person` `dbo:OfficeHolder` assigned through the DBpedia type hierarchy, for which we would choose the most specific type to include in our training data. The 108 yago[13] type categories assigned to him vary from `Actor` to `BodyBuilder` to `Emigrant` and `Traveler`. Whilst this does provide a richer representation of the entity, it may also introduce additional noise in the training dataset as the majority of `dbo:OfficeHolders` are not (former) body builders or actors.

If our approach is applied to running text instead of Wikipedia, the system could apply different types per document or mention. Nevertheless, it is better to use clear cases with single types for training and allow multiple types for applying/testing, either per data set or per document/mention.

Coverage of Types

The type information in the Spanish DBpedia dataset is less complete than for the Dutch and English DBpedias. This holds in particular if we look at the number of DBpedia types associated with each instance. Upon analysing the instance_type dumps from DBpedia 2016-04, we find that only 80.92% of the Spanish DBpedia instances have a DBpedia type associated with them versus 96.70% of the Dutch DBpedia instances. The coverage of the latter is similar to the English DBpedia (95.72%). But both Spanish and Dutch DBpedia datasets have fewer types associated with them than the English DBpedia. If we compare for example the English http://dbpedia.org/resource/San_Francisco,

[11] `dbpedia:` is shorthand for http://dbpedia.org/resource.

[12] `dbo:` is shorthand for http://dbpedia.org/ontology/.

[13] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/.

then there are in total 45 types assigned, of which 30 come from yago, 4 from umbel, 2 from schema.org, 2 from wikidata, 1 from w3c Basic Geo Vocabulary and 6 from the DBpedia ontology (including the top level node `owl:Thing`). The Dutch http://nl.dbpedia.org/resource/San_Francisco has 11 types associated with it in total, of which 7 are from the DBpedia ontology, 2 from schema.org types and two from wikidata. The Spanish http://es.dbpedia.org/resource/San_Francisco_(California) has nine types associated with it, 4 from the DBpedia hierarchy, 2 from schema.org, 1 from ontologydesignpatterns and one from Open Geospatial Consortium. The resource http://es.dbpedia.org/resource/Europa only has `skos:Concept` associated with it as its type. This limits the number of different type hierarchies that can be exploited for training a type classifier and the number of instances that can be included. However, we are looking into leveraging the type information in the English DBpedia and its links to other language DBpedias to experiment with different type hierarchies.

6 Conclusion and Future Work

We have presented a system and experiments for fine-grained entity typing for Dutch and Spanish. We show that our system performs comparable to systems presented previously for English. Furthermore, our approach is easily extensible to other languages for which Wikipedia and DBpedias exist. The trained models, as well as the code to extend the approach to other languages are available from GitHub.

Whilst using DBpedia as the typing system is a strength of the system in that it provides us with high quality typing information, it also has its limitations, in particular when certain types are not or insufficiently referenced in the dataset. We observe that the DBpedia types are better covered in the English DBpedia (which is no surprise as the type hierarchy was originally developed for English) and we intend to leverage this to provide DBpedia types to Dutch and Spanish entities that lack these.

Furthermore, we found that certain entity types used in prior work such as `car` and `person/education` have no equivalent in the DBpedia hierarchy. We therefore aim to experiment with different entity type hierarchies, whilst still preserving our 'clean' approach to generating training samples. This is particular interesting for domain-specific applications in which certain types that are less well defined in the current DBpedia hierarchy are more important.

Looking further, we aim to develop our system so that it can expand to hundreds of entity types in tens of languages.

Acknowledgements. The research for this paper was made possible by the CLARIAH-CORE project financed by NWO.

Appendix A: Results

Table 5. Precision, recall and F_1 scores on the overall datasets (macro-averaged) and per class.

Type	Dutch				Spanish			
	P	R	F_1	# instances	P	R	F_1	# instances
location/celestial	.99	.98	.98	1,895	.98	.93	.95	690
location/city	.98	.99	.99	179,919	.95	.98	.96	35,036
location/country	.99	.99	.99	142,240	.99	.99	.99	79,091
location/geography/island	.93	.84	.88	5,310	.94	.92	.93	3,473
location/geography/mountain	.97	.90	.93	2,940	.92	.83	.87	832
location/park	.94	.87	.91	253	.94	.89	.91	453
location/structure	.92	.88	.90	1,693	.92	.86	.89	1148
location/structure/airport	.96	.87	.91	575	.97	.85	.91	459
location/structure/hotel	.86	.83	.84	46	.79	.71	.75	31
location/structure/restaurant	.83	.42	.56	60	-	-	-	-
location/structure/sports_facility	.96	.72	.83	36	-	-	-	-
location/transit/bridge	.95	.92	.94	460	.97	.89	.93	149
location/transit/railway	.96	.96	.96	2,100	-	-	-	-
location/transit/road	.99	.98	.99	2,392	.93	.97	.95	286
organization	.98	.96	.97	1,895	.96	.96	.96	13,308
organization/company	.97	.96	.97	9,529	.94	.80	.86	55
organization/company/broadcast	.99	.99	.99	1,975	-	-	-	-
organization/company/news	.98	.98	.98	1,316	.97	.97	.97	2,058
organization/government	.97	.98	.97	419	-	-	-	-
organization/military	.98	.97	.97	1,025	-	-	-	-
organization/political_party	.98	.97	.97	2,714	.98	.97	.98	5,149
organization/sports_league	.99	.99	.99	1,485	-	-	-	-
organization/sports_team	.97	.87	.92	39	-	-	-	-
organization/transit	.97	.94	.96	1,285	-	-	-	-
other/art	-	-	-	-	.63	.70	.66	90
other/art/broadcast	.95	.93	.94	6,071	.91	.88	.89	5,972
other/art/film	.91	.86	.89	4,697	.81	.79	.80	4,637
other/art/music	.87	.66	.75	304	1.00	.56	.71	27
other/art/stage	.50	.20	.29	5	.88	.75	.81	838
other/art/writing	1.00	.91	.95	57	-	-	-	-
other/award	.93	.97	.95	70	.97	.97	.97	1,321
other/body_part	.95	.96	.96	1,298	-	-	-	-
other/currency	.98	.96	.97	898	.98	.91	.94	583
other/event	.92	.87	.90	305	-	-	-	-
other/event/election	.97	.98	.98	1,092	-	-	-	-
other/event/holiday	-	-	-	-	.94	.93	.94	665
other/event/protest	.97	.93	.95	61	-	-	-	-
other/event/sports_event	1.00	.98	.99	1,493	-	-	-	-
other/event/violent_conflict	.99	.98	.99	15,128	.97	.96	.97	9,238
other/health/malady	.98	.98	.98	4,964	.95	.97	.96	4,197
other/heritage	.90	.85	.88	2,167	-	-	-	-

(*continued*)

Table 5. (*continued*)

Type	Dutch				Spanish			
	P	R	F_1	# instances	P	R	F_1	# instances
other/internet	.97	.94	.95	891	.99	.97	.98	1,693
other/language	.98	.98	.98	25,781	.96	.96	.96	12,227
other/language/programming_ language	-	-	-	-	.91	.84	.88	186
other/legal	.99	.92	.95	148	-	-	-	-
other/living_thing/animal	-	-	-	-	.95	.89	.92	650
other/product	.96	.89	.93	76	-	-	-	-
other/product/software	.96	.94	.95	1,881	.91	.88	.89	1,022
other/religion	1.00	.99	.99	1,491	-	-	-	-
other/product/weapon	-	-	-	-	.92	.83	.87	471
other/scientific	.94	.90	.92	3,012	.90	.86	.88	4,512
other/sports_and_leisure	.99	.99	.99	6,696	1.00	1.00	1.00	5,082
other/supernatural	.89	.86	.87	1,947	.88	.87	.87	4,497
person	.90	.90	.90	10,821	.86	.86	.86	14,818
person/artist	.93	.88	.90	3,712	.90	.81	.85	2,723
person/artist/actor	.87	.87	.87	6,985	.89	.92	.90	18,428
person/artist/author	.93	.93	.93	6,538	.92	.89	.91	7,534
person/artist/music	.91	.93	.92	15,308	.91	.92	.92	24,234
person/athlete	.77	.73	.75	2,961	.80	.76	.78	6,544
person/doctor	1.00	.60	.75	10	-	-	-	-
person/legal	.31	.31	.31	36	-	-	-	-
person/political_figure	.89	.87	.88	5,088	-	-	-	-
person/title	.96	.94	.95	4,500	-	-	-	-
person/religious_leader	-	-	-	-	.96	.92	.94	2,030
Total	.94	.87	.90	498,355	.92	.88	.90	276,437

References

1. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the web of data. Web Semant. Sci. Serv. Agents World Wide Web **7**(3), 154–165 (2009)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Technical report, Archiv (2016). https://arxiv.org/abs/1607.04606
3. Corro, L.D., Abujabal, A., Gemulla, R., Weikum, G.: FINET: context-aware fine-grained named entity typing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015, pp. 868–878 (2015)
4. Ekbal, A., Sourjikova, E., Frank, A., Ponzetto, S.P.: Assessing the challenge of fine-grained named entity recognition and classification. In: Proceedings of the 2010 Named Entities Workshop at ACL 2010, Uppsala, Sweden, July 2010, pp. 93–101 (2010)
5. Gillick, D., Lazic, N., Ganchev, K., Kirchner, J., Huynh, D.: Context-dependent fine-grained entity type tagging. arXiv (2014)
6. Giuliano, C.: Fine-grained classification of named entities exploiting latent semantic Kernels. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CNLL, Boulder, Colorado, USA, pp. 201–209 (2009)

7. Hovy, D.: How well can we learn interpretable entity types from text? In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short papers), Baltimore, Maryland, USA, 23–25 June 2014, pp. 482–487. Association for Computational Linguistics (2014)
8. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. Technical report, arXiv (2016). https://arxiv.org/abs/1607.01759
9. Ling, X., Weld, D.S.: Fine-grained entity recognition. In: AAAI (2012)
10. Linguistic Data Consortium: ACE (automatic content extraction) english annotation guidelines for entities. Technical report, Linguistic Data Consortium, version 5.6.6 2006.08.01 (2006)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
12. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3–26 (2007)
13. Nadeau, D., Turney, P.D., Matwin, S.: Unsupervised named-entity recognition: generating gazetteers and resolving ambiguity. In: Lamontagne, L., Marchand, M. (eds.) AI 2006. LNCS, vol. 4013, pp. 266–277. Springer, Heidelberg (2006). doi:10.1007/11766247_23
14. Nakashole, N., Tylenda, T., Weikum, G.: Fine-grained semantic typing of emerging entities. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–9 August 2013, pp. 1488–1497. Association for Computational Linguistics (2013)
15. Nothman, J., Curran, J., Murphy, T.: Transforming wikipedia into named entity training data. In: Proceedings of the Australasian Language Technology Association Workshop, pp. 124–132 (2008)
16. Recasens, M., de Marneffe, M.C., Potts, C.: The life and death of discourse entities: identifying singleton mentions. In: Proceedings of NAACL (2013)
17. Ren, X., He, W., Qu, M., Hang, L., Ji, H., Han, J.: AFET: automatic fine-grained entity typing by hierarchical partial-label embedding. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), Austin, TX, USA, 1–5 November 2016
18. Sekine, S., Sudo, K., Nobata, C.: Extended named entity hierarchy. In: LREC (2002)
19. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, pp. 142–147. Association for Computational Linguistics (2003)
20. Weischedel, R., Hovy, E., Marcus, M., Palmer, M., Belvin, R., Pradhan, S., Ramshaw, L., Xue, N.: Ontonotes: a large training corpus for enhanced processing. In: Olive, J., Christianson, C., McCary, J. (eds.) Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation, pp. 54–63. Springer, New York (2011)
21. Yaghoobzadeh, Y., Schütze, H.: Corpus-level fine-grained entity typing using contextual information. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015, pp. 715–725. Association for Computational Linguistics (2015)
22. Yaghoobzadeh, Y., Schütze, H.: Multi-level representations for fine-grained typing of knowledge base entities. In: Proceedings of the European Chapter of the Association for Computational Linguistics (EACL), 3–7 April 2017. https://arxiv.org/abs/1701.02025 (2017, to appear)

23. Yao, L., Riedel, S., McCallum, A.: Collective cross-document relation extraction without labelled data. In: Proceedings of EMNLP (2010)
24. Yogatama, D., Gillick, D., Lazic, N.: Embedding methods for fine grained entity type classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015), Short papers, Bejing, China, 26–31 July 2015, pp. 291–296. Association for Computational Linguistics (2015)
25. Yosef, M.A., Bauer, S., Hoffart, J., Spaniol, M., Weikum, G.: HYENA: hierarchical types classification for entity names. In: Proceedings of COLING 2012: Posters, Mumbai, India, December 2012, pp. 1361–1370 (2012)