Multi-word Entity Classification in a Highly Multilingual Environment

Sophie Chesney[*,†], Guillaume Jacquet[†], Ralf Steinberger[†] and Jakub Piskorski[†]

[†]Text and Data Mining Unit, Joint Research Centre, European Commission, Ispra, Italy
[*]Cognitive Science Research Group, School of Electronic Engineering and Computer Science,
Queen Mary University of London, UK
s.chesney@qmul.ac.uk
{firstname.lastname}@jrc.ec.europa.eu

Abstract

This paper describes an approach for the classification of millions of existing multi-word entities (MWEntities), such as organisation or event names, into thirteen category types, based only on the tokens they contain. In order to classify our very large in-house collection of multilingual MWEntities into an application-oriented set of entity categories, we trained and tested distantly-supervised classifiers in 43 languages based on MWEntities extracted from BabelNet. The best-performing classifier was the multi-class SVM using a TF.IDF-weighted data representation. Interestingly, one unique classifier trained on a mix of all languages consistently performed better than classifiers trained for individual languages, reaching an averaged F1-value of 88.8%. In this paper, we present the training and test data, including a human evaluation of its accuracy, describe the methods used to train the classifiers, and discuss the results.

1 Introduction

Named Entities (NEs) such as persons, organisations, locations or events are crucial bearers of information as they are often the answers to major text understanding questions. Software to carry out Named Entity Recognition (NER) in free text needs to recognise the relevant strings in text and disambiguate the broad entity types (e.g. *Paris Hilton* is a person rather than a location), justifying the term Named Entity Recognition and Classification (NERC). In this paper we focus on MWEntity classification, thereby placing NERC in the context of the study of MWExpressions.

Our work is carried out in a highly multilingual environment, and as a result, suitable training corpora are difficult to source. Motivated by this, in addition to a method of MWEntity classification, we also present a technique for automatically generating a silver-standard annotated resource of 3.8 million entities for use as training data. This resource incorporates data from 43 different languages, covering multiple language families. MWEntities are often not translated, so it is rather common to find names from one language in amongst entities from another (e.g. French MWEntity 'institut polytechnique des sciences avancées' found in the Arabic dataset).

It is important to specify that our classification work is exclusively based on internal features of the names; that is, the tokens contained within each MWEntity. No additional external features were extracted. This is due in part to the fact that the contexts of our historically accumulated MWEntities are no longer known. We therefore aim at developing a system that can be widely applied to data sets that do not include, or give access to, such contextual information.

The paper begins with a section on related work (2) and is followed by a section describing the starting point and the objective of our work: the target entity hierarchy (3.1); the set of entities extracted from the BabelNet resource (Navigli and Ponzetto, 2012) and the method used for the extraction (3.2); and an evaluation of this BabelNet silver-standard including inter-annotator agreement data (3.3). In Section 4, we present the classification methods we tested, i.e. a baseline approach and two variants of Support Vector Machines. Experiments and results achieved are presented in Section 5, together with a discussion of the results. We conclude with a short summary and a pointer to future work.

Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017), pages 11–20,
Valencia, Spain, April 4. ©2017 Association for Computational Linguistics

2 Related Work

In this task, we work exclusively on the classification of MWEntities, which are subject to their own idiosyncrasies and difficulties. Though many of the papers discussed below do not necessarily exclude multi-word units in their NERC systems, none of them explicitly focus on MWEntities. Furthermore, although a large body of work exists on the study of multi-word expressions more generally, including idioms (Villada Moirón and Tiedemann, 2006; Gharbieh et al., 2016), fixed expressions such as 'in short' and compound nominals such as 'car park', work focusing exclusively on multi-word named entities is less prominent in the literature. Here, we are interested in this subset of MWExpressions in the task of Named Entity Classification (NEC), particularly as they tend to be richer in word-internal features, upon which our systems are based.

Early NERC systems began emerging during the 1990s, favouring handcrafted rule-based approaches. Due to the fact that these systems offer control over results and straight-forward fine-tuning, many industrial NERC systems continue to be rule-based, at least to some extent (Steinberger, 2012). In an academic context, however, machine learning approaches to automatically detecting such rules have become more popular in recent work. The majority of recent NERC systems use supervised learning, relying on large, often manually annotated corpora from which to extract and learn positive and negative features for a particular class of entity. Since such corpora are costly, attention has also turned to distant-supervision, which utilises existing structured resources (e.g. WordNet (Fellbaum, 1998), DBPedia (Auer et al., 2007), Freebase (Bollacker et al., 2008), BabelNet (Navigli and Ponzetto, 2012), among others) to automatically generate 'silver-standard' annotated corpora, without incurring the cost associated with gaining access to manually annotated corpora (e.g. Fleischman and Hovy (2002), Ling and Weld (2012), Nothman et al. (2013)). We follow this general approach with the production of a large-scale automatically-created MWEntity resource extracted from BabelNet, used to distantly supervise our classifiers. Similarly, weakly-supervised systems use a bootstrapping technique to approach the same issue, starting with a few annotated examples and automatically expanding the corpus based on these 'seed' terms (e.g. Pasca et al. (2006), Ratinov and Roth (2009)).

In this work, we are interested in drawing a distinction between the recognition of named entities and, most relevant to us, their classification. The task of entity classification has been approached largely through machine learning techniques, utilising both word-internal features (Durrett and Klein, 2014) and additional contextual information, such as dictionary definitions (Gangemi et al., 2012) and 'lexical expansions' (e.g. synonyms and derivationally related forms) extracted from WordNet, as well as co-occurrence statistics from external corpora (Del Corro et al., 2015).

Very recent work has also moved towards multi-source learning, automatically retrieving additional semi-structured contextual information, such as webpage titles and URLs, through Web search (Vexler and Minkov, 2016).

Much of the early work in the area of NERC was monolingual, often working on English data. As approaches have advanced, multilingual named entities have received more attention, though the reliance on large corpora often limits the possible coverage. In an attempt to overcome this bottleneck, Nothman et al. (2013) automatically classify Wikipedia articles into named entity types, exploiting the links between in-text entities and their corresponding Wikipedia pages. The authors therefore engineer a silver-standard annotated corpus of named entities in nine languages (English, German, French, Polish, Italian, Spanish, Dutch, Portuguese and Russian), for use as training data for NERC systems. In this work, we approach large-scale multi-word entity classification in 43 languages, developing a highly multilingual NE classification system tailored specifically for MWEntities, using distant-supervision.

3 Extraction of a Multi-Word Entity Silver-Standard Resource from BabelNet

When addressing the MWEntity recognition task, some approaches are based on methods that make the classification of recognised MWEntities difficult. This is the case for approaches using co-occurrences of MWEntities and their acronyms (Jacquet et al., 2016), or those derived from n-gram methods (Ekbal and Saha, 2013). In both cases, the method is able to extract MWEntities from text and consider them as one expression, but cannot provide an entity type for these expressions. Also, although many publicly available entity resources exist, they often are difficult to use in a specific application for a variety of reasons. For example, the provided entity types may not correspond to what is required for the specific application, may be too specific or too coarse-grained, or not provided at all. In these cases, there is a strong need to (re-)annotate an existing resource of MWEntities. To address this goal, we propose a method of creating a silver-standard data set from BabelNet. We defined the required annotation types for our specific application and extracted the entities and their variants which have the hypernyms corresponding to these annotation types from BabelNet. We conducted a partial manual evaluation of the obtained resource, discussed in Section 3.3.

3.1 Named Entity Type Hierarchy

Related to Sekine's (2002) Extended Named Entity (ENE) Hierarchy[1], our own in-house entity hierarchy contains nine major classes (person, organisation, location, event, product, identifier, time, number and Other) with altogether almost fifty sub-classes.

In our existing text processing system, many of these NE categories are already recognised and classified (e.g. persons, cities, email addresses, date expressions), so these are not considered here. In this paper, we focus on classifying MWEntities according to a subset of thirteen categories shown in Table 1, corresponding to the types requiring more fine-grained annotation in our system.

ORGANISATION		
Subtype	**Example**	**Encoding**
POLITICAL-PUBLIC	Democratic Party	ORG-PP
COMMERCIAL	Microsoft Inc.	ORG-CO
SPORT	FC Barcelona	ORG-SP
EDUC-RESEARCH	University of Lugano	ORG-ER
LOCATION		
Subtype	**Example**	**Encoding**
FACILITY	Schiphol Airport	LOC-FA
OTHER	Mount Everest	LOC-OT
PRODUCT		
Subtype	**Example**	**Encoding**
ELECTRONICS	Commodore 64	PRO-EL
WEAPON	AGM-1 Carbine	PRO-WE
VEHICLE	Mitsubishi Pajero	PRO-VE
ART	Star Wars	PRO-AR
EVENT		
Subtype	**Example**	**Encoding**
INCIDENT	Chernobyl Disaster	EVT-IN
NATURAL	Hurricane Katrina	EVT-NA
OCCASION	Nobel Prize Awards	EVT-OC

Table 1: Types used for NE-classification task.

3.2 Automatically-Created Annotated Resource from BabelNet

For the sake of creating resources for each of the named-entity types listed in Table 1, we have exploited BabelNet (Navigli and Ponzetto, 2012), a large multilingual encyclopaedic dictionary and semantic network, created by merging various publicly available linguistic resources, e.g. WordNet and Wikipedia. BabelNet contains circa 7.7 million NE-related synsets. In order to extract sought-after entities, we used the BabelNet API[2]. Since the NE-related BabelNet synsets are not tagged with a specific NE tag, the NE type was inferred by using the hypernym information provided in BabelNet (i.e. using WordNet hypernyms and Wikipedia categories). To be more precise, based on hypernym frequency information for the entire set of named entities contained in BabelNet, for each NE type a list of *positive* and *negative* hypernyms was manually created. These lists were subsequently used to extract entities of each particular type. A given NE-related synset was extracted if: (a) there was at least one hypernym for the main sense of the synset in the list of positive hypernyms, and (b) no hypernym for the main sense of the synset was on the list of negative hypernyms. For instance, the full list of positive and negative hypernyms for extracting commercial organisation names (ORG-CO) is given in Table 2.

positive hypernyms	**negative hypernyms**
company, periodical, magazine, record_company, publisher, airline, enterprise, corporation, bank, brewery, automobile_manufacturer, film_production_company, limited_company, joint-stock_company, holding_company telephone_company, drug_company, investment_company, shipping_company, oil_company, electric_company, train_operating_company, telecommunication_company, bank_holding_company, consulting_company, moving_company, transport_company, consultancy, factory, private_bank	city, City, settlement, town, metropolis, municipality, village, commune, park, capital, earthquake, tsunami, fire, avalanche, hurricane, flood, port, mountain, person

Table 2: The list of positive and negative hypernyms for the extraction of commercial organisation names (ORG-CO).

The main drive behind the usage of a negative hypernym list was to filter out potentially ambiguous named entity candidates, e.g. the same name might refer to a person, organisation and a loca-

[1] http://nlp.cs.nyu.edu/ene/version7_1_0Beng.html

[2] http://babelnet.org/guide

tion. The list of positive/negative hypernyms for each of the 13 categories varied. However, no list contained more than 100 items.

In total, we obtained circa 3.8 million named entities from BabelNet after expanding each extracted NE-related synset. The left-hand columns in Table 3 provide a breakdown of the number of extracted entities per type.

Entity Type	#Extracted Entities	#Filtered Entities
ORG-PP	214 056	100 373
ORG-CO	440 522	158 502
ORG-SP	285 312	139 578
ORG-ER	271 486	144 137
LOC-FA	1 182 857	469 633
LOC-OT	782 578	207 053
PRO-EL	33 053	8 817
PRO-WE	29 044	10 238
PRO-VE	55 494	17 617
PRO-AR	363 356	141 541
EVT-IN	68 647	38 139
EVT-NA	14 292	7 920
EVT-OC	94 908	54 256
TOTAL	3 835 605	1 497 804

Table 3: Number of entities extracted from BabelNet before and after filtering (see Section 5.2).

MWEntity	Ref annot.	Manual annot.
Examples of full agreement (167 MWEntities over 200)		
lisnagarvey high school	ORG-ER	ORG-ER
teeside mohawks	ORG-SP	ORG-SP
grand château dansembourg	LOC-FA	LOC-FA
a writers nightmare	PRO-AR	PRO-AR
maritsa hotel	LOC-FA	LOC-FA
slaughter grüning and company	ORG-CO	ORG-CO
At least 3 different annotations from 4 annotators (8 MWEntities over 200)		
vic urban	ORG-CO	...
st marys badley	LOC-FA	...
go gaia	ORG-CO	...
tarnobrzeg voivodship	ORG-PP	...
rez quad	ORG-ER	...
janet jeffrey carlile harris carillon	LOC-FA	...
lindley court	ORG-ER	...
the church on brady	LOC-FA	...
All annotators agreed, but disagreed with ref. (6 MWEntities over 200)		
colt revolver	ORG-CO	PRO-WE
rip mountain	LOC-FA	LOC-OT
accademia florence	ORG-ER	LOC-FA
childrens champion awards	ORG-CO	EVT-OC
1999 nato bombing of valjevo	ORG-CO	EVT-IN
buffalo rochester and pittsburgh railroad	ORG-CO	LOC-FA

Table 4: Some example of MWEntities to be annotated.

3.3 Manual Evaluation of the Automatically-Created Resource from BabelNet

A crucial element of our work consisted of evaluating the quality of the automatically generated annotated resource from BabelNet. To justify its use as a gold (or 'silver') standard resource for this supervised classification task, we conducted a small manual evaluation, shown in Table 5, with native speakers of five different languages (German, French, Polish, English and Swedish), evaluating both the quality of the automatic annotations as well as inter-annotator agreement for English across four annotators (one of whom is a native English speaker).

Each annotator was trained on a trial set of 100 randomly extracted English MWEntities, then tested on a further 200 randomly extracted multiword entities for their own native language, and an additional 200 for English. The annotators were asked to provide two separate sets of annotations: first, the annotators provided 'offline' annotations for each of the entities, selecting from a set of 13 possible entity types (corresponding to the types described in Table 1). The no-guess tag ('NG') was used when an annotation decision could not be made with certainty, or when an entity appeared to belong in a category not included in the possible list of tags. Secondly, the annotators were permitted to research their secondary guess 'online'. For consistency, the BabelNet labels were hidden throughout.

Table 5 shows that the 'offline' annotation results are quite heterogeneous among annotators, with a precision between 81.6% and 92.4%, and a recall between 66.5% and 81.0%. On the other hand, the 'online' annotation results are much more homogeneous, for the same language between different annotators, and also across languages: precision varied between 87.6% and 92.5%, and recall between 85.0% and 90.5%. The averaged kappa across the 4 English annotators is 0.848, and among the 200 annotated MWEntities, 159 were annotated with full inter-annotator agreement, including only 6 which differed from the automatically-generated BabelNet annotation (listed in Table 4). 10 were annotated with the same type by 3 of 4 annotators. The remaining 31 MWEntities, where only two annotators agreed,

Languages	'offline' annotation			'online' annotation			SVM_tfidf (lang. indep.)		
	P	R	F1	P	R	F1	P	R	F1
ENGLISH									
a1 (Nat.)	83.9%	75.5%	79.5%	92.5%	86.5%	89.4%	87.5%	87.5%	87.5%
a2	92.4%	66.5%	77.3%	91.3%	89.0%	90.1%			
a3	86.7%	72.0%	78.7%	88.7%	86.5%	87.6%			
a4	82.5%	80.0%	81.2%	91.1%	87.5%	89.3%			
FRENCH	89.9%	80.5%	85.0%	91.9%	90.5%	91.2%	91.5%	91.5%	91.5%
POLISH	81.6%	75.5%	78.4%	90.7%	87.5%	89.1%	85.5%	85.5%	85.5%
GERMAN	83.2%	77.0%	80.0%	87.6%	85.0%	86.3%	84.0%	84.0%	84.0%
SWEDISH	86.6%	81.0%	83.7%	90.7%	87.5%	89.1%	77.0%	77.0%	77.0%

Table 5: Manual annotations on 200 MWEntities randomly extracted for 5 languages from the created resource, compared with the best-performing system (right-most column).

highlight the difficulty of the task: some MWEntities are ambiguous, and could easily be annotated with different types. For example, *'buffalo rochester and pittsburgh railroad'* could be annotated both as a company or a facility. This manual evaluation aims to show that, although the resource we extracted from BabelNet is not perfect, it is consistent enough across annotators and languages to consider it a silver-standard in our experiments.

4 MWEntity Classification Approaches

We present two main approaches to the multi-class MWEntity classification problem described above: a baseline using cosine similarity, and two variations of (distantly) supervised Support Vector Machines. We use Scikit-learn (Pedregosa et al., 2011), the machine learning library for Python, for implementing the different approaches.

4.1 Baseline Approach: CosSim

The baseline approach adopted in this classification task, hereafter referred to as the CosSim system, is modelled on a simple search engine, where query word vectors are compared with document vectors through cosine similarity. In our case, a query word vector is analogous to the expression to be classified, while the document vectors are analogous to vectors representing each category in the training set. The type associated with the category vector most similar to the query vector is selected as the classification for the query expression. For each category, using a TF.IDF vectorisation process, we generate a ranking in the importance of terms that can be considered a type of 'topic signature' (Fleischman and Hovy, 2002) for this category, since words more strongly associated with a particular category receive higher TF.IDF scores. When no token in the to-be-classified multi-word entity occurs in the training data for a given category, this expression will receive a cosine similarity score of 0 with this category. If this is the case for all categories, CosSim is unable to classify the expression and instead outputs a no-guess label ('NG'). Both the training and test expressions are vectorised with a TF.IDF vectoriser (Pedregosa et al., 2011), with standard L2 normalisation (to normalise for variation in the number of expressions found in each category) and sublinear TF calculations (which log-scales the TF counts).

4.2 SVM Approaches: SVM_tfidf & SVM_counts

We develop two supervised Support Vector Machine (SVM) classifiers which differ only in the vectorisation method adopted: SVM_tfidf utilises the same TF.IDF vectoriser as CosSim, while SVM_counts uses a simple count vectoriser. We therefore follow a simple bag-of-words (BoW) model for extracting TF.IDF and count-based features from the tokens contained within each MWEntity. Classification is 'pairwise (One-Versus-One; OVO), meaning that a binary classifier is trained for each pair of classes and the class which receives most votes (highest count) is selected. This method of multi-class classification was favoured over One-Versus-Rest classification to minimise training time, following Hsu and Lin (2002). This is implemented using Scikit-learn's LinearSVC SVM classifier with the One-Versus-One wrapper (Pedregosa et al., 2011). We chose an SVM classification approach following its widely-acknowledged strong performance on text classification tasks (Joachims, 1998; Yang and Liu, 1999; Qin and Wang, 2009; Ye et al., 2009).

4.3 Confidence Thresholds

We were interested in whether we could utilise the scores of the CosSim, SVM_tfidf, and SVM_counts as parameters for maximising for precision or recall in the classification task, as this is particularly relevant in the context of our specific application. We therefore define 5 threshold levels corresponding to the lower percentiles of the scores at 5% intervals (0, 5, 10, 15 and 20%) in order to evaluate whether this method has the desired effect, and calculate the exact score thresholds using `numpy.percentile()`[3]. For each classification with a confidence or similarity score below the threshold, the expression in question is re-classified with the no-guess tag ('NG').

Both SVM systems *always* attempt to classify an expression, so at the 0% threshold there will be no 'NG' classifications; however, as detailed in Section 4.1, CosSim does not classify an expression if it has a similarity score of 0 with all possible categories, instead classifying with 'NG' also at the 0% threshold.

5 Evaluations

This section provides a brief discussion of the method of cross-validation used in this work and an overview of the preprocessing carried out on the resource automatically generated from BabelNet, before turning to the experimental method and results of the experiments.

5.1 Cross-Validation

The automatically annotated resource from BabelNet is separated into 43 languages, varying in coverage. We use 10-fold shuffle-split cross-validation, split 75% training and 25% testing for all experiments detailed below. The general approach was as follows (any discrepancies from this will be explicitly detailed later where necessary): the data for each language is randomly shuffled (with a constant random state initialisation value for reproducibility) 10 times, and each shuffled version is then separated for training and testing. With this method, it is not guaranteed that each fold will be different, but it is likely with sizeable data sets; nonetheless, we favour this technique over k-fold cross-validation as it maximises the training data available, even for the smallest languages in the resource.

5.2 Preprocessing

When preparing the automatically generated resource from BabelNet for use in the MWEntity classification task, we considered only those entities that consist of at least two tokens, and additionally removed some potentially problematic entries (i.e. entities containing only two tokens including one with a single character).

In addition, we excluded non-alphanumerical strings and removed all duplicates within each entity category. We did not exclude entities which occurred in more than one category, as we argue that removing such cases would lead to a bias in the results.

Following this method of filtering, approximately 1.5 million entries were retained for the experiments. Table 3 provides a breakdown of the initial number of extracted entities per type and the final number of entities that were used for the purpose of carrying out the MWEntity classification experiments.

We also experimented with replacing all numerical characters with the same token ('0'), after observing that certain classes contain many similarly formatted numerical tokens, such as dates. In these tests, we chose to replace each number character individually, in order to retain some distinctions between classes: for example, taken from the Swedish data set, 'EVT-NA' contains a large number of dates ('2004 asiatiske tsunami' → '0000 asiatiske tsunami'), while 'PRO-WE' contains mixed alphanumerical strings ('mp40 schmeisser' → 'mp00 schmeisser'). Despite the intuition that replacing numerical characters in this way would create more generalised features for the classes in question, this in fact had little or no positive effect in the classification task using the SVM_tfidf method, and a significantly negative effect with the CosSim method. Consequently, it was not adopted in the full-scale experiments described below.

5.3 Experiments

During development, we compared the performance of the two SVM systems, SVM_tfidf and SVM_counts. In line with the expectation that TF.IDF vectorisation would provide more informative features in the task of differentiating between categories, we found SVM_tfidf performed marginally better overall. In the following full-scale experiments, we therefore will only dis-

[3] `http://www.numpy.org/`

Excluded	Language dependent			Language independent		
percentile	P	R	F1	P	R	F1
CosSim						
0%	81.8%	61.5%	66.3%	81.3%	62.8%	67.0%
5%	83.2%	59.8%	65.0%	82.8%	59.9%	65.1%
10%	84.1%	56.7%	63.1%	83.7%	56.6%	62.5%
15%	85.1%	53.6%	60.6%	84.1%	53.4%	59.7%
20%	85.9%	50.4%	58.0%	85.0%	50.2%	56.9%
SVM_TFIDF						
0%	87.8%	87.5%	87.5%	**88.9%**	**88.8%**	**88.8%**
5%	90.0%	85.4%	87.4%	**91.6%**	**86.6%**	**88.6%**
10%	91.8%	82.6%	86.6%	**92.6%**	**83.3%**	**87.5%**
15%	93.0%	79.0%	84.9%	**93.4%**	**79.5%**	**85.5%**
20%	93.6%	75.0%	82.8%	**94.2%**	**75.4%**	**83.3%**

Table 6: Average results across the 43 tested languages, with language-dependent or independent approaches, for the 5 tested percentile thresholds.

cuss the comparison between SVM_TFIDF and the baseline approach, CosSim.

The main task compared the performance of language-dependent and language-independent training for the two classification methods, when applied across 43 languages at 5 different threshold levels (see Section 4.3 for threshold definitions). The 43 languages correspond mostly to European languages including Russian, plus Arabic.

5.3.1 Language-Dependent Training

For each of the classification methods, SVM_TFIDF and CosSim, a language-specific classifier is built for each of the 43 languages in the resource. Using the method of 10-fold cross-validation described in Section 5.1, the data for each language is separated for training and testing, to allow for a language-by-language comparison on the performance of each classification method. We compare the performance of SVM_TFIDF with the baseline CosSim across each of the 5 threshold levels for all 43 languages.

5.3.2 Language-Independent Training

In order to fairly compare the performance of a language-independent classifier with those with language-dependent training, testing is still carried out language-dependently (we use the same test sets in both experiments). We create a language-independent training corpus by concatenating each of the language-specific training sets from the previous experiment, and importantly, excluding any duplicate MWEntities, so as to remove any overlap between training and testing data. Once again, we compare the performance of SVM_TFIDF with the baseline CosSim across each of the 5 threshold levels for all 43 languages.

5.4 Results

As Table 6 shows, in both experiments, SVM_TFIDF is the best-performing classification method in precision, recall and F1, across all percentile thresholds. The baseline, CosSim, performs marginally worse in terms of precision, but significantly worse in terms of recall (across all thresholds and both training methods). We exclude a higher percentage of low-scoring classifications in the threshold experiments, leading to a distinct improvement in precision, in the best case increasing by over 5.7% in SVM_TFIDF. This supports the intuition that the scores assigned by both the SVM systems and the CosSim system correlate to the accuracy of the chosen label. This can therefore be viewed as a means of tweaking or prioritising precision over recall or vice versa. Maximising precision with the 20% threshold, when averaged across all languages, we achieve precision of over 94% with the language-independent SVM_TFIDF classifier. More specifically, we see precision of over 95% in 16 of the 43 languages tested.

The best system overall is the language-independent SVM_TFIDF classifier, significantly outperforming the CosSim system when trained on both language-dependent and independent data, especially in terms of recall and F1. At its peak, we see a marked difference of over 26% between the two systems in terms of recall, both trained on language-independent data. We also see consistent improvements in precision, recall and F1 across all percentile thresholds from SVM_TFIDF trained on language-dependent to independent data.

In particular, Table 7 shows a significant boost

Selected languages	Support	Description	Language dependent P	 R	 F1	Language independent P	 R	 F1
Romanian	23588	best	94%	94%	94%	96%	96%	96%
English	713656	largest	88%	88%	88%	88%	88%	88%
Faroese	120	smallest/worst	56%	55%	50%	74%	68%	67%
Arabic	16520	non-Latin	87%	87%	87%	88%	88%	88%
Russian	43936	non-Latin	86%	85%	85%	87%	87%	87%

Table 7: Results for some specific languages of the 43 tested, with language-dependent or independent approaches, with SVM_TFIDF method at the 0% percentile threshold.

Selected classes	Support	Description	Language dependent P	 R	 F1	Language independent P	 R	 F1
EVT-NA	7920	best	96.4%	91.4%	93.6%	96.8%	94.4%	95.6%
LOC-FA	469633	largest	89.9%	93.4%	91.7%	90.6%	94.4%	92.1%
ORG-CO	158502	worst prec.	76.9%	80.6%	78.5%	81.3%	81.4%	81.4%
PRO-EL	8817	worst recall	83.5%	58.6%	67.8%	82.8%	65.3%	72.8%

Table 8: Results for some specific type classes, with language-dependent or independent approaches, with SVM_TFIDF method at the 0% percentile threshold.

in performance in Faroese, the smallest language in the data set (from an F1 of 50% to 67%), with little or no impact on English, the largest portion of the resource. Similar improvements are seen in the other under-represented languages in the resource: Ladino (F1 67% to 88%), Luxembourgish (F1 77% to 88%) and, to a lesser extent, Romansh (F1 81% to 82%). This suggests that utilising cross-linguistic data to supplement the training data for the smaller languages is beneficial.

At the 0% percentile threshold, the language achieving the best results is Romanian, with precision, recall and F1 well above the 88% average for this system, at 96%. Furthermore, we also see minor improvements on languages not using the Latin alphabet, such as Arabic and Russian, suggesting that language-independent training can even improve performance in cases where we would expect that language-specific features would be most useful. This is likely due to the fact that a single-language corpus often contains some portion of international terms.

Table 8 shows that language-independent training also causes a small boost in performance across individual class types. In particular, a marked improvement is made in the 'PRO-EL' class, which achieves the worst recall value with language-dependent training, improving by 6.7%. In general, Table 8 demonstrates that performance varies across classes, with a particularly striking difference in recall between the best-performing class ('EVT-NA') and the worst ('PRO-EL'). Given that these two classes are relatively close in size, this suggests class size is not the unique driving factor in performance and that different NE categories are linguistically diverse.

6 Conclusion and Future Work

We presented an approach to automatically classify MWEntities based only on their internal features. We described how to construct a silver-standard resource of MWEntities from BabelNet adapted to an application-driven type hierarchy. The classifiers were applied in a highly multilingual environment, 43 languages, and we showed how they perform better when trained on all languages combined, with a language-independent training set. With the SVM_TFIDF approach, using 10-fold shuffle-split cross-validation on a 1.5 million MWEntity data set, we obtained a precision/recall of 88.9%/88.8% when all expressions are classified, and 94.2%/75.4% when we filter the 20% least confident classifications. We also showed that these results are reasonably stable across languages, being more sensitive to the number of expressions available to train this language than to its scripting. In addition, we demonstrated that, despite the fuzzy delimitation between entity types, for instance between facilities and organisations, the classifiers perform reasonably well for all entity types.

We now plan to explore one more method: using the best-performing classifier (training all languages combined using SVM and a TF.IDF-weighted data representation) on character n-grams. We hope that this may help to capture words that are similar across languages, but not identical (e.g. *national / nazionale / nacional / nationaal*).

We will then apply the best-performing classifier to our vast and equally highly multilingual in-house collections of MWEntities. As our in-house collections also contain MWExpressions that are not entities (e.g. *Chief Executive Officer*, *kilometres per hour*), we will have to face the challenge of having to identify the expressions that are not covered by the classes we have trained. We hope that the thresholds to exclude the least confident classifications will be efficient at that task.

7 Acknowledgement

This work was carried out while the first author was a member of the Text and Data Mining Unit of the Joint Research Centre, European Commission. The author is now based in the Cognitive Science Research Group at Queen Mary University of London.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 722–735, Berlin, Heidelberg. Springer-Verlag.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. Finet: Context-aware fine-grained named entity typing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 868–878. Association for Computational Linguistics.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association of Computational Linguistics*, 2:477–490.

Asif Ekbal and Sriparna Saha. 2013. Stacked ensemble coupled with feature selection for biomedical entity extraction. *Knowledge-Based Systems*, 46:22–32.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Michael Fleischman and Eduard Hovy. 2002. Fine grained classification of named entities. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Aldo Gangemi, Andrea Giovanni Nuzzolese, Valentina Presutti, Francesco Draicchio, Alberto Musetti, and Paolo Ciancarini. 2012. Automatic typing of dbpedia entities. In *International Semantic Web Conference*, pages 65–81. Springer.

Waseem Gharbieh, Virendra Bhavsar, and Paul Cook, 2016. *Proceedings of the 12th Workshop on Multiword Expressions*, chapter A Word Embedding Approach to Identifying Verb-Noun Idiomatic Combinations, pages 112–118. Association for Computational Linguistics.

Chih-Wei Hsu and Chih-Jen Lin. 2002. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425.

Guillaume Jacquet, Maud Ehrmann, Ralf Steinberger, and Jaakko Väyrynen. 2016. Cross-lingual linking of multi-word entities and their corresponding acronyms. In *Proceedings of the 10th Language Resources and Evaluation Conference*, Portorož, Slovenia.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *AAAI*.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. 2013. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175.

Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Organizing and searching the world wide web of facts-step one: the one-million fact extraction challenge. In *AAAI*, volume 6, pages 1400–1405.

Fabian Pedregosa, Gal Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David, Cournapeau, Matthieu Brucher, Matthieu Perrot, and douard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Yu-ping Qin and Xiu-kun Wang. 2009. Study on multi-label text classification based on svm. In *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on*, volume 1, pages 300–304. IEEE.

Lev Ratinov and Dan Roth, 2009. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, chapter Design Challenges and Misconceptions in Named Entity Recognition, pages 147–155. Association for Computational Linguistics.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended named entity hierarchy. In *LREC*.

Ralf Steinberger. 2012. A survey of methods to ease the development of highly multilingual text mining applications. *Language Resources and Evaluation*, 46(2):155–176.

Reuth Vexler and Einat Minkov, 2016. *Proceedings of the Sixth Named Entity Workshop*, chapter Multi-source named entity typing for social media, pages 11–20. Association for Computational Linguistics.

Begoña Villada Moirón and Jörg Tiedemann, 2006. *Proceedings of the Workshop on Multi-word-expressions in a multilingual context*, chapter Identifying idiomatic expressions using automatic word-alignment.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM.

Qiang Ye, Ziqiong Zhang, and Rob Law. 2009. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications*, 36(3):6527–6535.