Natural Language Engineering **23** (1): 31–51. © Cambridge University Press 2015
doi:10.1017/S1351324915000431

A classification approach for detecting cross-lingual biomedical term translations

H. HAKAMI[1] and D. BOLLEGALA[2]
[1]*Computer Science Department, Taif University, Saudi Arabia*
e-mail: hoda.h@tu.edu.sa
[2]*Department of Computer Science, The University of Liverpool, UK*
e-mail: danushka.bollegala@liverpool.ac.uk

(*Received 31 January 2015; revised 2 November 2015; accepted 3 November 2015; first published online 14 December 2015*)

Abstract

Finding translations for technical terms is an important problem in machine translation. In particular, in highly specialized domains such as biology or medicine, it is difficult to find bilingual experts to annotate sufficient cross-lingual texts in order to train machine translation systems. Moreover, new terms are constantly being generated in the biomedical community, which makes it difficult to keep the translation dictionaries up to date for all language pairs of interest. Given a biomedical term in one language (source language), we propose a method for detecting its translations in a different language (target language). Specifically, we train a binary classifier to determine whether two biomedical terms written in two languages are translations. Training such a classifier is often complicated due to the lack of common features between the source and target languages. We propose several feature space concatenation methods to successfully overcome this problem. Moreover, we study the effectiveness of contextual and character *n*-gram features for detecting term translations. Experiments conducted using a standard dataset for biomedical term translation show that the proposed method outperforms several competitive baseline methods in terms of mean average precision and top-*k* translation accuracy.

1 Introduction

Technical terms such as biomedical terms are constantly being generated and need to be correctly translated into numerous languages in order to facilitate the flow of knowledge in a technical community. Although comprehensive monolingual lexicons of biomedical terms are created for the English language, only a fraction of those terms are translated into other languages. For example, the Unified Medical Language System Metathesaurus, one of the comprehensive multilingual medical resources covering twenty-one languages, contains 75.1 per cent English terms, 9.99 per cent Spanish terms, 2.22 per cent Japanese terms, 1.82 per cent French terms, and only 10.87 per cent for terms in all other languages.[1] The contrastingly disproportionate representation of languages other than English in the Unified

[1] http://nlm.nih.gov/research/umls

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

Medical Language System demonstrates the necessity to invent methods that can automatically detect translations for English biomedical terms in other languages. Considering that technical fields such as biomedicine are continuously growing, it is both labour-intensive and costly to manually translate all the novel biomedical terms into all target languages. Even a more critical factor is the lack of bilingual experts that can participate in such a large-scale manual annotation process. Therefore, it is necessary to invent methods that can automatically detect translations across biomedical lexicons.

Having been provided with two lists of biomedical terms corresponding to a source and a target language, we propose a method to detect pairs of terms that are translations between the two languages. First, we represent a term using two types of features: *character n-grams* extracted from the term under consideration, and *contextual features* consisting of words that appear within a contextual window surrounding the term under consideration. Character n-grams can be considered as *intrinsic* features because they are extracted from the term itself and represent syllables and inflections in a language. On the other hand, contextual features can be considered as *extrinsic* features as they represent the other words that are frequently associated with the term under consideration. For example, ('dwarfism', 'nanisme') is pair of English–French terms that has ('dw', 'wa',. . . , 'ism') as English character n-gram features and ('na', 'an',. . . , 'sme') as French character n-gram features. Regarding the contextual features, the most frequent words that appear around ***dwarfism*** term are *condition*, *syndrome*,. . . , etc and for ***nanisme*** the words such as *atteindre* and *origine* appear frequently in its context. The two types of features are mutually exclusive by design. Therefore, by combining the two feature spaces we can potentially better represent the properties of a term in a language.

We train a binary classifier using the two feature types described above. Specifically, we use a training dataset of term pairs (w_S, w_T) where the two terms w_S and w_T are selected respectively from the source S and the target T languages, and w_T is the translation of w_S. However, naive concatenation of feature vectors representing each term is insufficient to represent the training instances because, the character n-gram and contextual features extracted from different languages do not necessarily overlap. Moreover, the exact method of weighting character n-gram and contextual features enabling them to be used within the same classifier is non-obvious.

We empirically compare different feature weighting and concatenation approaches for this purpose. Specifically, we consider four basic feature spaces in our model: source-language n-gram feature space (S_n), source-language contextual feature space (S_c), target-language n-gram feature space (T_n), and target-language contextual feature space (T_c). We generate both first-order and second-order combinations of those feature spaces resulting in six different cross-lingual feature spaces. The exact procedures for generating feature spaces are detailed in Section 3.1.4. To our knowledge, such exhaustive analysis of cross-lingual feature spaces has not been conducted before, which can be considered as an important novel contribution of this work.

The classifier returns a confidence score indicating the probability that two terms are translations. Given a term w_S, we use the trained classifier to compare w_S with each of the terms w_T in the target biomedical term list and rank w_T according to

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

their probability of being a correct translation for w_S. This is particularly useful for human annotators when compiling bilingual term dictionaries as it reduces the effort from going through a large list of terms to identify the correct translations to a few top-ranked candidates.

We evaluate our proposed method on the English–French language pair. Experiments have been conducted for character *n*-grams and contextual features separately and also their incorporation. Our experiments show that using character *n*-gram features to measure the probability score of a pair of terms to be a translation achieves more accurate results in terms of mean average precision (MAP) than using contextual features. The proposed method achieves MAP scores of respectively eighty-one per cent and sixty-three per cent for character *n*-grams features and contextual features. Moreover, the combination of character *n*-grams and contextual features produces a MAP score of eighty-six per cent, which indicates that we can achieve more effective performance by combining the two feature spaces as opposed to when they are used separately.

We first discuss relevant prior related work in Section 2. Next, in Section 3, we explain the proposed methods for learning similarity of terms across languages. We present experimental results in Section 4 and then conclude the paper.

2 Related work

Character *n*-grams extracted from a technical term provide useful etymological and inflectional information about that term. Prior works in machine translation have used character *n*-grams as features for detecting translations of words (Mcnamee and Mayfield 2004; Namer and Baud 2005; Vilar, Peter and Ney 2007; Xi *et al.* 2012). Vilar *et al.* (2007) treat a word as a sequence of characters to establish the translation model of words between two related languages that have corresponding structures such as Catalan–Spanish and Spanish–Portuguese. Their experimental results show that character-based translation is possible between languages that use similar alphabets such as English and French. Character-level translation models have been particularly effective for learning translation models between resource poor languages (Tiedemann 2012; Tiedemann and Nakov 2013). Combining word-level and character-level features has shown to further improve the accuracy of these models (Nakov and Tiedemann 2012).

Earlier work on building bilingual dictionaries using machine-learning approaches with character *n*-gram features has shown encouraging results (Claveau 2008; Erdmann *et al.* 2009; Kontonatsios *et al.* 2014b). Kontonatsios *et al.* (2014b) propose a novel method to recognize semantically equivalent biomedical terms in language pairs using a random forest (RF) classifier (Breiman 2001). They exploit the internal structure of sequences using the character *n*-grams to align terms across languages. Their method performs robustly on two language pairs: English–French and English–Chinese. Erdmann *et al.* (2009) train a support vector machine classifier using labelled term-translation pairs for the purpose of automatically constructing a bilingual dictionary. They experimentally show that the trained classifier can predict the correctness of unseen term-translation pairs with high accuracy. In bilingual

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

information retrieval, Mcnamee and Mayfield (2004) found that using character *n*-gram models is highly effective even for not closely related languages.

On the other hand, distributional similarity has been widely used in NLP for several purposes such as discovering the semantic relationships between biomedical terminologies to organize them into groups (Fan and Friedman 2007; Weeds *et al.* 2007). A distributional semantic model has been successfully applied in order to automatically discover pairs of semantically related words in large monolingual text corpora (Rapp 2008; Dias *et al.* 2010). However, those works focus on applying distribution similarity models in a single language. In contrast, cross-lingual distributional similarity models compare the distributional similarity for technical terms across pairs of languages using a context vector to find the target words that have the most similar distributions for a given source word (Rapp 1999; Chiao and Zweigenbaum 2002; Saralegi, San Vicente, and Gurrutxaga 2008; Kontonatsios *et al.* 2014b). In the methods developed by Rapp (1999), Chiao and Zweigenbaum (2002) and Saralegi *et al.* (2008), a context vector is transferred from a source word into target language's target context vector relying on existing bilingual lexicon in order to make the vectors comparable. Then, a similarity score is computed between translated context vectors with each target word context vector to produce a ranked list of candidate translations. Similarly, Kontonatsios *et al.* (2014b) use the contextual model to compile a bilingual dictionary of technical terms from English–Spanish comparable corpora. They observed that the character *n*-gram model significantly outperformed the contextual model in terms of top-*k* translation accuracy.

To take advantage of lexical composition and distributional similarity, Kontonatsios et al. 2014a developed a hybrid method that combines compositional and contextual similarity scores as features in a linear classifier. By investigating the performances using top-1 and top-20 translation accuracy, they show that combining those two different scores improves the top-20 translation accuracy, whereas minor improvements are observed for the top-1 accuracy. Unlike our proposed method, they use a bilingual seed dictionary to map the context vector between the two languages in order to make them comparable using similarity measures such as the cosine similarity. Finally, both of the scores obtained by using the character *n*-grams-based model and the context vectors are combined as features to train a linear classifier.

Translating contextual features with the usage of a dictionary arises several issues. First, a competent bilingual dictionary is required as the seed dictionary. Often a small amount of translations is insufficient to translate a large contextual feature space. Therefore, both the quality and the coverage of the translation dictionary directly affects the performance of the term translation method. Second, not all contextual features (e.g. bigrams) are proper phrases in a language. Therefore, only a fraction of the contextual features could be correctly translated using a bilingual dictionary. On the other hand, the main purpose of our work is to build a wide-coverage bilingual dictionary. Therefore, it would be unrealistic to assume the availability of a dictionary to translate contextual feature spaces. In our proposed method, we train a classifier using contextual features of source and target languages without translating contexts from source language to the target language.

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

3 Measuring cross-lingual term similarity

In this section, we describe our proposed method for measuring the similarity between source and target language terms for detecting translations. We model the problem of detecting term translation as a binary classification task. First, we describe the features that we use to represent words in a language. Second, we describe the binary classifier we train for detecting whether two words are translations. The confidence that the trained classifier has about two terms being translations is considered as the degree of cross-lingual similarity between those terms.

3.1 Feature engineering

Designing features for a classifier is an integral part of the machine-learning pipeline. We represent a term in a language using two types of features. Our first feature type is character *n*-grams extracted from a term that we would like to represent. Character *n*-gram features capture useful properties about a term such as its inflections. For example, the trigram prefix *mal* as in *malnutrition* often indicates the deficiency of a substance. Character *n*-gram features are described in detail in Section 3.1.1. The second feature type we use to represent a term is contextual features – words that appear within a certain contextual window surrounding the term under consideration. According to the distributional hypothesis, words that are semantically similar occur in similar contexts. Therefore, if a source term and a target term co-occur with similar contexts, then it is likely that those two terms are translations. The contextual features we extract for representing terms are described in Section 3.1.2. Character *n*-gram features and contextual features are complementary in the sense that character *n*-gram features are extracted from the term under consideration and contextual features are extracted from the contexts in which the term appears. Consequently, we concatenate the two feature types to create a hybrid feature space as described later in Section 3.1.3.

Given a feature vector for a source term and a target term, which are in a translational relationship, we must construct a single feature vector to represent the training instance consisting of the two terms. We can then train a binary classifier using the pairs of terms in which the two terms are translations of each other (i.e. positive training instances) and not (i.e. negative training instances). However, it is non-obvious as how to construct a single feature vector that captures information from the two feature vectors representing the source and the target terms. For this purpose, we compare two methods: *first-order* feature combinations, and *second-order* feature combinations, which we detail in Section 3.1.4.

3.1.1 Character n-gram feature space

Character *n*-grams are consecutive sequences of characters in the order of their appearance in source/target terms, where *n* corresponds to the number of characters in each sequence. The Examples of English and French character *n*-gram features are presented in Table 1. Multi-word terms often demonstrate a compositional structure. Therefore, character *n*-grams are likely to capture semantic units such as inflections

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

Table 1. *Example of character n-grams features for English biomedical term antibody, and its French translation anticorps*

English–French training instance	(antibody, anticorps)
English n-grams features (S_n)	an, nt, ib, bo, od, dy, ant, nti, tib, ibo, bod, ody
French n-grams features (T_n)	an, nt, ti, ic, co, or, rp, ps, ant, nti, tic, cor, orp, rps
Feature vector	[EN+an, EN+nt, EN+ib, …, FR+cor, FR+orp, FR+rps]

Table 2. *English Contextual features representing English biomedical terms alopecia, abnormality, and acetaldehyde. The corresponding PPMI values are shown in the table*

EN features EN terms	Disease	Patient	Treatment	Infection	Blood
Alopecia	0	32.847	19.665	0	0
Abnormality	199.344	126.353	80.565	101.189	124.721
Acetaldehyde	0	0	0.573	0	11.707

or etymological components in large terms. The number of n-grams that can be generated from a set of terms grows exponentially with the length of the n-gram. Consequently, we consider n-grams for $n = 2, 3, 4$, and 5 in this work, and select the most frequent n-grams to overcome this n-gram explosion problem. We denote the n-gram feature space for the source and target languages respectively by S_n and T_n.

3.1.2 Contextual feature space

The contexts in which a term occurs provide useful hints regarding the semantics of that term (Baroni and Lenci 2010). In fact much prior work in lexical semantics such as similarity measurement (Lin 1998), or word sense disambiguation (Chan and Ng 2005) exploit this contextual information. We extract unigrams and bigrams as *contextual features* from the contexts in which a term occurs in a corpus in order to represent that term. We use a window of seven words surrounding the term under consideration as the context of that term for extracting its contextual features. Prior work on vector space models of semantics (Turney and Pantel 2010) have shown that it is sufficient to consider window sizes in the range from five to ten tokens for capturing the local context of a word. We denote the source and target language contextual feature spaces respectively by S_c and T_c. Tables 2 and 3 respectively show a subset of English (EN) and French (FR) contextual features. Each row in Table 2 has its translation term represented in the corresponding row in Table 3. A prefix that identifies whether it originated from the source or the target language and whether it is a character n-gram or a contextual feature is assigned to each feature.

Weighting co-occurrences between a term and its contextual features is important because contextual features that do not often co-occur with a term are unlikely to be salient features representing that term. We use positive pointwise mutual information, which has shown excellent performance in numerous NLP tasks (Turney

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

Table 3. *French Contextual features representing French biomedical terms calvitie, malformation, and acétaldéhyde. The corresponding PPMI values are shown in the table*

FR feature FR terms	Pouvoir	Maladie	Traitement	Faire	Devoir
Calvitie	18.734	0	0	0	0
Malformation	115.113	84.158	2.030	0.078	32.788
Acétaldéhyde	51.286	0	0	0	0

and Pantel 2010) for this purpose. Pointwise mutual information takes into account the frequency of source term and source language contextual features (similarly for target terms). Pointwise mutual information takes values in the range $[0, +\infty)$ and is calculated as follows:

$$\text{PPMI}(w_1, w_2) = \max\left(0, \log\left(\frac{P(w_1, w_2)}{P(w_1)P(w_2)}\right)\right), \tag{1}$$

where $P(w_1, w_2)$ is the joint probability of the two words w_1 and w_2, $P(w_1)$ and $P(w_2)$ are respectively marginal probabilities of w_1 and w_2. All probabilities are estimated using corpus counts.

3.1.3 Hybrid models

Character *n*-gram features are extracted directly from a term under consideration, whereas contextual features are extracted from the contexts in which that term occurs. Therefore, the two feature spaces are mutually exclusive and capture different aspects of a term. To incorporate both feature types, we propose a hybrid feature space by concatenating the two feature spaces into a single feature space. The block diagram shown in Figure 1 illustrates the proposed hybrid model, which includes character *n*-gram and contextual features in the same classifier. During training, each training instance, given as a pair of source and target terms, is represented by the concatenated vector of the four feature vectors: (a) source term's character *n*-gram feature vector, (b) source term's contextual feature vector, (c) target term's character *n*-grams feature vector, and (d) target term's contextual feature vector. We consider further variants for constructing a feature vector to represent a training instance in Section 3.1.4.

3.1.4 First and second-order feature spaces

We model the problem of detecting whether a given target language term w_T is the translation of a source language term w_S, as a binary classification task. Specifically, we classify term-pairs (w_S, w_T) as positive or negative indicating whether w_T is respectively a correct translation or an incorrect translation of w_S. We describe the exact learning algorithm later in Section 3.2 and describe the procedure we propose to represent a pair of terms (w_S, w_T) using a feature vector $f(w_S, w_T)$.

As we already described in Sections 3.1.1 and 3.1.2, we use two types of features, giving rise to four feature vectors representing the two terms w_S and w_T as follows:

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

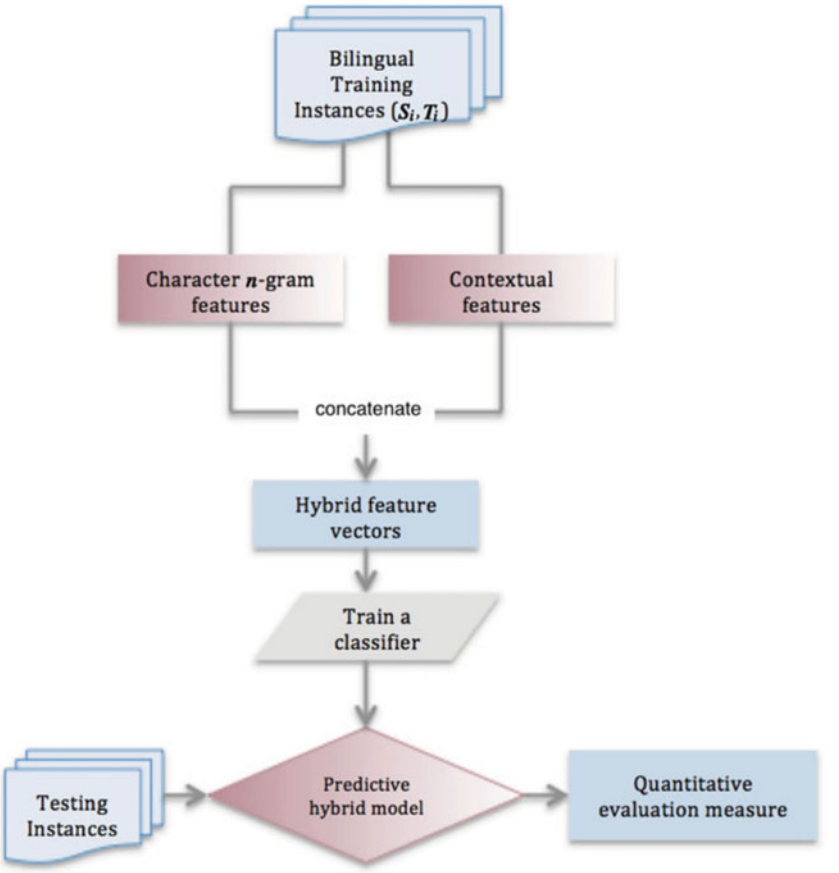


Fig. 1. (Colour online) Integrating character n-gram and contextual features into the same classification model (hybrid model).

- $f(w_S, S_n)$: Source language character n-gram feature vector for the source term;
- $f(w_S, S_c)$: Source language contextual feature vector for the source term;
- $f(w_T, T_n)$: Target language character n-gram feature vector for the target term;
- $f(w_T, T_c)$: Target language contextual feature vector for the target term.

We propose two methods for creating a single feature vector to represent a term-pair (w_S, w_T), given the above-mentioned four feature vectors.

First, we consider linear concatenation of character n-gram, contextual, and hybrid feature spaces to construct three flavors of feature vectors as follows:

- $S_n + T_n$: The concatenation of source and target character n-grams feature vectors $f(w_S, S_n)$ and $f(w_T, T_n)$;
- $S_c + T_c$: The concatenation of source and target contextual feature vectors $f(w_S, S_c)$ and $f(w_T, T_c)$;

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

- $S_n+S_c+T_n+T_c$: The concatenation of source and target character *n*-grams and contextual features (hybrid model), which corresponds to the concatenation of the four vectors $f(w_S, S_n)$, $f(w_S, S_c)$, $f(w_T, T_n)$, and $f(w_T, T_c)$.

The linear concatenation of an *n*-dimensional vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^\top$, and an *m*-dimensional vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)^\top$ is defined as the $(n + m)$ dimensional vector $(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m)^\top$. We refer to the feature spaces created by the linear concatenation of feature vectors for w_S and w_T as *first-order feature vectors.*

Although first-order feature vectors consider the features for both source and target language terms, they do not consider the correlation between features across languages or feature spaces. To overcome this problem, we propose *second-order feature vectors* that are constructed considering all pair-wise combinations of features. Specifically, given an *n*-dimensional vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^\top$, and an *m*-dimensional vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)^\top$, their second-order combination is given by the *nm*-dimensional vector $(x_1y_1, x_1y_2, \ldots, x_ny_m)^\top$. Numerous different combinations can be considered to create second-order features, however we have extracted and evaluated the following in-combination second-order features:

- $S_n \otimes T_n$: The pairwise combination of the source *n*-gram feature vector and target *n*-gram feature vector;
- $S_c \otimes T_c$: The pairwise combination of the source contextual feature vector and target contextual feature vector;
- $(S_n \otimes T_n) + (S_c \otimes T_c)$: The concatenation of the above two feature vectors.

In all pairwise combinations, we multiply the two corresponding feature values to compute the feature value of the second-order feature. Multiplication operator has been found to be useful in compositional semantics models in vector spaces (Mitchell and Lapata 2008). Indeed, also in our case multiplication is more efficient than addition because unless a feature is present both in the source and the target vectors, there will be a zero second-order feature when using multiplication. We consider multiword terms and hyphenated terms as single lexical units for the purpose of character *n*-gram extraction and contextual feature extraction.

3.2 Translation detection as a binary classification problem

We model the problem of determining whether a target term w_T is the correct translation of a given source term w_S as a binary classification problem. Specifically, given a pair of words (w_S, w_T), we train a binary classifier that returns $+1$ if w_T is the correct translation of w_S, and -1 otherwise. For training, we assume the availability of correct target language translations for a set of source language terms. We create positive training instances by coupling each source language term with its correct target language translation. We generate an equal number of negative training instances by randomly pairing a source language term with a target language term. We use the *first-order* and *second-order* feature vectors that we described in Section 3.1.4 to represent a pair of terms. During test time, the trained binary classification model can be used to predict the probability of a given target language term w_T being the correct translation of a source language term w_S. Specifically,

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

the class conditional probability $P(+1|w_S, w_T)$ is used to rank the target language translation candidates w_T for a source language term w_S.

Having considered two binary classification algorithms that provide class conditional probabilities: logistic regression classifier (LR), and RF (Breiman 2001), we briefly overview each of those classification algorithms. We use the implementations in the scikit-learn Python library for those two algorithms in our experiments.[2] The RF is an ensemble-learning mechanism in which multiple decision trees are learnt for collectively predicting the class of an instance. Similar to the decision tree algorithms, RF can build complex combinations of features to learn classification rules from training data. Several methods have been proposed in the literature for learning RF with class probability outputs (Boström 2007). Moreover, RF has proved to be effective for very high dimensional feature spaces such as the feature spaces used in our experiments, where the number of features exceeds the number of the training instances (Díaz-Uriarte and De Andres 2006).

As a log-linear classifier that produces probabilistic outputs, we train a binary LR. We use l_2 regularization, with the regularization coefficient determined by cross-validation, in order to reduce the overfitting to train data. Unlike the RF classifiers that consider combinations of features, the LR classifier learns a weight for each feature in the train instances that indicates the discriminative power of that feature when separating positive training instances from the negative ones.

We use stratified five-fold cross validation method with MAP, described later in Section 4.2.1, to tune classifier's parameters. Each fold uses twenty per cent of the train data for validation, and the remainder of eighty per cent for training. In stratified cross-validation, the folds are made by preserving the percentage of samples for each class. In the case of the RF classifier, the number of trees in the forest influences the classifier's performance. In our preliminary experiments, we observed that the optimal number of trees is different for RF depending on the feature space. These differences could be explained by the fact that the *first-order* feature spaces require more decision trees to capture the association among source and target features compared to *second-order* bilingual features because, in the *second-order* feature spaces, we have already generated pairwise combinations of features. Therefore, we conclude that the number of trees required to boost the performance of a classifier depends on the training feature space. For this reason, we automatically adjust this parameter for each experiment before building a model for training.

One of the main parameters to adjust when using the regularized LR classifier is the l_2 regularization coefficient c. In theory, smaller values for c result in a higher regularization, that is, they increase the complexity of the model (i.e. model with non zero features). According to our findings, this parameter affects the models differently depending on the feature space. For example, the most appropriate choice of regularization coefficient for $(S_n \otimes T_n)$feature space is 1,000, whereas c equal to 10 results in a more accurate model for $(S_n \otimes S_n)$ and $(S_c \otimes T_c)$ feature spaces. Because of

[2] `http://scikit-learn.org`

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

Table 4. *Dataset statistics*

	Train data	Test data
English Terms	3,530	760
French Terms	3,530	9,090
Ratio of English multi-word terms	0.82	0.21
Ratio of French multi-word terms	0.76	0.24
Ratio of English hyphenated terms	0.02	0.007
Ratio of French hyphenated terms	0.09	0.262
Wikipedia English corpus size (in tokens)	1.9M	4.8M
Wikipedia French corpus size (in tokens)	1.1M	2.2M

those differences, we applied the *stratified k-fold* cross validation to each experiment with the purpose of identifying the best value of a parameter for each feature space.

4 Experiments

4.1 Dataset

For training a classifier, we use a bilingual dictionary of English and French biomedical terms. We use the dataset described in Kontonatsios *et al.* (2014a) to train and evaluate our proposed method. First, a set of 9,090 English biomedical terms with one or more French translations listed is manually selected from the Unified Medical Language System metathesaurus. The selected terms cover disease names, drug names, and chemical substances. The terms are selected without restricting to any particular biomedical subdomain, and can be considered as a broad collection of terms covering different subdomains. We refer to this term translation pairs as the *seed dictionary*.

Kontonatsios *et al.* (2014a) created a comparable biomedical corpora using the Wikipedia medical articles as a source. Having selected 4,000 English biomedical terms from the Unified Medical Language System metathesaurus, they selected the Wikipedia articles whose titles contain one or more terms from the set of selected biomedical terms. Then, they follow the Wikipedia interlingual links to retrieve the semantically related articles in the target language. Although the accuracy of the information collected from cloud sourced encyclopedic resources such as Wikipedia is debatable, the issue is complementary to the translation prediction problem we consider in this paper. The collected comparable corpora is further divided into train and test parts for extracting contextual features. Table 4 summarizes the statistics of the data used in our experiments.

We generate positive training instances for training the binary classifiers (LR and RF) by pairing each English term with one of its French translations. We generate negative training instances by pairing an English term with a randomly chosen French term. We verify that the generated negative training instance term-pairs are not correct translations by comparing them against the seed dictionary. The technique of randomly sampling pseudo-negative instances has been successfully used

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

in several tasks in NLP such as noise contrastive estimation for word representation learning (Mikolov, Chen and Dean 2013a; Mikolov, tau Yih and Zweig 2013b; Bollegala, Maehara and ichi Kawarabayashi 2015), and binary classifier training in semantic similarity measurement (Bollegala, Matsuo and Ishizuka 2007). Although the technique is suboptimal compared to manually creating negative instances, it is popularly applied due to the fact that it obviates the manual effort required in data annotation.

In total, our training dataset consists of 9,090 positive and 9,090 negative training instances. The train dataset is denoted by $\{((S_1, T_1), y_1), ((S_2, T_2), y_2), \ldots, ((S_n, T_n), y_n)\}$, where $S_1, S_2, \ldots, S_n$ stand for the feature vectors of source terms, $T_1, T_2, \ldots, T_n$ stand for the feature vectors of target terms and $y_1, y_2, \ldots, y_n$ stand for labels. A positive training sample is defined as a pair of source and target terms (S_i, T_i), where T_i is the correct translation of S_i. Positive instances are assigned a label of $+1$, whereas a negative instance a label of -1. A feature vector is created to represent a training instance by concatenating the two feature vectors representing the source and target terms. In addition to the seed dictionary we use for training, we select 1,000 English terms and their French translations as test data. We verified that there is no overlap between the sets of terms selected for train and test purposes.

4.2 Measuring the performance

4.2.1 Mean average precision

In our proposed system, the ranked probabilities of target language candidates are returned in the descending order. An efficient term similarity prediction method across languages must assign a higher probability score to the correct target term translation for a given source term. Therefore, if our learned model ranks the correct target language (French) translation as high as possible, then our method could be considered as effective. This can be evaluated using MAP, which takes into account the order in which the returned target terms are presented. MAP is a strict measure in the sense that it not only considers whether the correct results are ranked among the top-k results but also takes into account whether those correct results are ranked as high as possible. The MAP for a set of source language (English) test terms is the MAP for each test term and is calculated as follows:

$$\mathrm{MAP} = \frac{\sum_{i=1}^{N} \mathrm{AveP}(i)}{N}. \tag{2}$$

Here, N stands for the total number of source language test terms and $AveP(i)$ is the average precision for i^{th} source test term, which is determined by the following formula:

$$\mathrm{AveP}(i) = \frac{\sum_{r=1}^{N} \mathrm{Precision}(r) \times \mathrm{Correct}(r)}{\text{No.of correct target translation in top } k}, \tag{3}$$

where Correct(r) is a binary-valued function that returns 1 if the target term at rank r is a correct translation for a given source term, otherwise it returns 0. $Precision(r)$

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

Table 5. *An example of calculating average precision for the translation candidates retrieved for the English term* mucosa. *Two correct French translations are ranked at the first and fourth places*

English test term (*mucosa*)					
FR-term	Muqueuse	Cause	Boucher	Muqueux	Vol
Rank	1	2	3	4	5
Correct/relevant	1	0	0	1	0
Precision	1/1	1/2	1/3	2/4	2/5
Average precision= (1/1+2/4)/2= 0.75					

is the precision at rank r, which is defined as

$$\text{Precision}(r) = \frac{\text{No.of correct target translations in top } r}{r}. \quad (4)$$

In our test dataset, most of the English test terms have only one correct translation. However, a small set has two or three correctly corresponding French terms. For example, English test term 'mucosa' has the correct French terms 'muqueux' and 'muqueuse', the system orders the French candidate as in Table 5 (top-5).

4.2.2 Top-k translation accuracy

In addition to MAP, we use top-k translation accuracy as an evaluation measure. Top-k accuracy is the percentage of source terms for which their correct translations were encountered among the top-k ranked target terms and is defined as follows:

$$\text{top-}k \text{ accuracy} = \frac{\text{no. of terms whose correct translation is below or equal to rank } k}{N} \times 100. \quad (5)$$

Prior works on biomedical term translation have used top-k accuracy as an evaluation measure (Rapp 1999; Chiao and Zweigenbaum 2002; Kontonatsios *et al.* 2014a). In this evaluation, we calculate the translation accuracy on the top-k of the returned ranked list for the target terms. For example, when translating English biomedical terms into French, top-10 translation accuracy indicates the percentage of English test terms for which at least one of the correct French translations is ranked among the top-10 ranked candidates. Top-k accuracy is a useful measure when the proposed method is used as a suggestion system for human translators because even if all of the correct translations are not ranked at the top, a human translator can benefit from a system that lists at least one correct translation among the top ranks so that he or she can select from that list. First, we determine the rank of the correct translation of each test term and then aggregate them, where N stands for the total number of English test terms. We compute the top-k accuracy for k values in the range from 1 to 20.

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

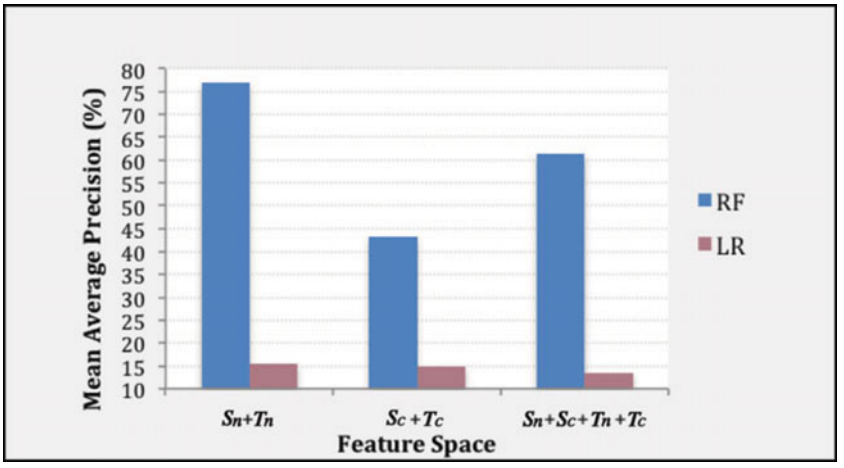


Fig. 2. (Colour online) MAP of the RF and LR classifiers on the first-order feature spaces.

High MAP values often indicate high top-k values, although the reverse is not always true. Therefore, MAP can be considered as a more sensitive evaluation measure than the top-k. However, we use top-k translation accuracy in addition to MAP so that we can with higher accuracy compare our results with the results reported in prior work that uses top-k as the only evaluation measure.

In the case of one classifier with specific feature space performing more accurately than the other one, it is important to determine if the two classifiers are statistically significantly different from each other. We use the binomial exact test with Clopper–Pearson confidence intervals (Clopper and Pearson 1934) in order to compare the performance obtained by different feature spaces and their combinations proposed in the paper. The Clopper–Perason confidence interval for a confidence level α can be computed as

$$\left[B\left(\frac{\alpha}{2}; x, n-x+1\right), B\left(1-\frac{\alpha}{2}; x+1, n-x\right)\right]. \tag{6}$$

Here, x stands for the number of correctly classified instances out of a total of n test instances. $B(\alpha; x, n)$ is the cumulative probability density function of the binomial distribution. If the intervals of two different classifiers do not overlap, it can be concluded that their performances are statistically significantly different.

4.3 Results

Figure 2 shows the MAP scores achieved by the RF and LR on first-order feature spaces. The RF shows good performance using first-order character n-gram and contextual features because the decision trees in the RF are able to automatically identify the associations among source and target features. In contrast, the LR performed poorly using the first-order n-gram or contextual features spaces. The LR is a log-linear classifier and does not combine the source and target first-order feature spaces when predicting whether a target term is a translation of a given source term.

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

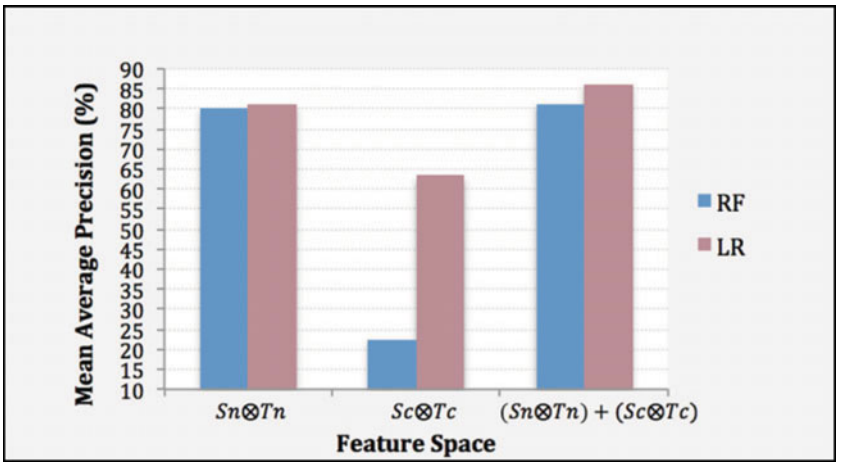


Fig. 3. (Colour online) MAP of the RF and LR classifiers on the second-order feature spaces.

When the RF is trained on first-order feature spaces, we observe that using character *n*-gram features alone in ranking the most similar target terms for a given source term results in more effective performance than contextual features and the integration of character *n*-gram and contextual features. In addition, the incorporation of first-order character *n*-grams and contextual features to train the RF classifier results in around eighteen per cent improvement compared to using only the contextual model.

Having implemented the RF and LR classifiers using second-order feature spaces (as shown in Figure 3), we observed that the second-order features significantly improved the performance of the (log-linear) LR classifier. The LR outperformed the RF by one per cent, forty per cent, and five per cent respectively in character *n*-gram, context and combined models. Experimental results demonstrate that the highest performance is achieved with the LR when integrating character *n*-gram and contextual second-order features (86.12 per cent), which gives a five per cent improvement over character *n*-gram alone, and approximately twenty-three per cent higher than contextual feature performance. All improvements reported are statistically significant according to the Clopper–Pearson confidence interval under $\alpha = 0.05$.

Due to the fact that the correct target translation of a source term is closely associated with the similar contextual features, the incorporation of contextual features enables the model to detect cross-lingual contextual similarity, thereby improving the classification accuracy. Moreover, for the RF, the highest performance was achieved with the hybrid model as opposed to using each feature space separately. What is also noticeable with the second-order feature space experiments is the fact that the accuracy of the RF classifier with contextual features dropped significantly (by twenty per cent). Table 6 summarizes the performance achieved by the LR and RF classifiers with all different feature spaces using MAP. It is noteworthy that the LR and RF achieved the highest performance with the hybrid model. Although it is difficult to compare the reported results in prior work

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

Table 6. *MAP scores obtained by the LR and RF classifiers for different feature spaces*

Features	Computation	LR	RF
Character *n*-grams	$S_n + T_n$	15.523	76.879
	$S_n \otimes T_n$	81.022	80.081
Context	$S_c + T_c$	14.986	43.264
	$S_c \otimes T_c$	63.285	22.455
Hybrid model	$S_n + S_c + T_n + T_c$	13.483	61.381
	$(S_n \otimes T_n) + (S_c \otimes T_c)$	**86.119**	**81.303**

on biomedical term translation due to of the differences in train/test data, the pre-processing methods used, and evaluation criteria that have been employed, we can consider the superiority of the second-order feature spaces proposed in this paper over the first-order feature spaces (consisting of linear combinations of character *n*-grams and contextual features) used in much prior work as implied by the superiority of the proposed method over prior work on this task.

Table 7 illustrates an interesting aspect of the probabilities assigned by the LR classifier between source and target test terms using each feature type separately as well as their combination. From Table 7, we see that character *n*-gram features are more reliable than contextual features. In other words, the LR mostly assigns comparatively higher probabilities to relevant terms when trained on character *n*-grams features as opposed to when done so using contextual features. This might be because that character *n*-gram features are useful for closely related languages such as English and French, as in our case. Another significant point is that probabilities show how incorporating those two types of features increases the probability of the relevant source/target term as well as decreases the probability of the irrelevant pairs.

Figure 4 illustrates the top-*k* translation accuracy for the LR classifier with character *n*-grams, contextual and hybrid models. For *k* values in the range from 1 to 20, the highest top-*k* accuracy is achieved with LR when a hybrid model has been learnt, with 69.68 per cent of English test terms having their translation listed at rank 1, and that being 88.26 per cent at rank 20. Experimental results for the character *n*-gram model show that about sixty-five per cent of the English test words have their correct translation as the first ranked word, and the correct translation is found among the top-20 candidates for 82.5 per cent of those terms. Therefore, using a hybrid model improves the accuracy of the LR by approximately four per cent for the top-1 results, and by six per cent for top-20 translation candidates. The lowest top-*k* translation accuracy of the LR is observed for contextual model in which the correct translation is ranked first for only 25.5 per cent of the test terms, and is found among the top-20 candidates for 76.5 per cent of those terms.

In Figure 5, top-1 to top-20 translation accuracy of the RF classifier is shown with the second-order character *n*-grams, contexts, and hybrid models. From top-1

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

Table 7. *Probabilities given by the LR classifier for pairs of test terms*

S term	*T* term	Rel(1) / Irrel(−1)	Pr(+1|(S_i, T_i)) Character *n*-gram	Context	Both features
Larva	Larve	1	0.981	0.867	0.999
	coexister	−1	0.281	0.540	0.080
Asthma	asthme	1	0.977	0.698	0.999
	asthmatique	1	0.832	0.861	0.990
	accueillir	−1	0.060	0.466	0.008
Secretion	sécrétion	1	0.982	0.855	0.999
	vertébral	−1	0.060	0.536	0.0096
Louse	pou	1	0.635	0.586	0.821
	glucide	−1	0.154	0.223	0.013
Regurgitation	régurgitation	1	0.985	0.543	0.999
	Relais	−1	0.113	0.411	0.005
Breathing	Respiration	1	0.825	0.906	0.990
	aortique	−1	0.202	0.639	0.077
Tear	larme	1	0.920	0.258	0.978
	acidocétose	−1	0.086	0.643	0.031
Heat	Chauffage	1	0.907	0.345	0.935
	ostáo-tendineux	−1	0.173	0.217	0.004
Angina	angor	1	0.867	0.792	0.989
	neurostimulation	−1	0.039	0.298	0.0008

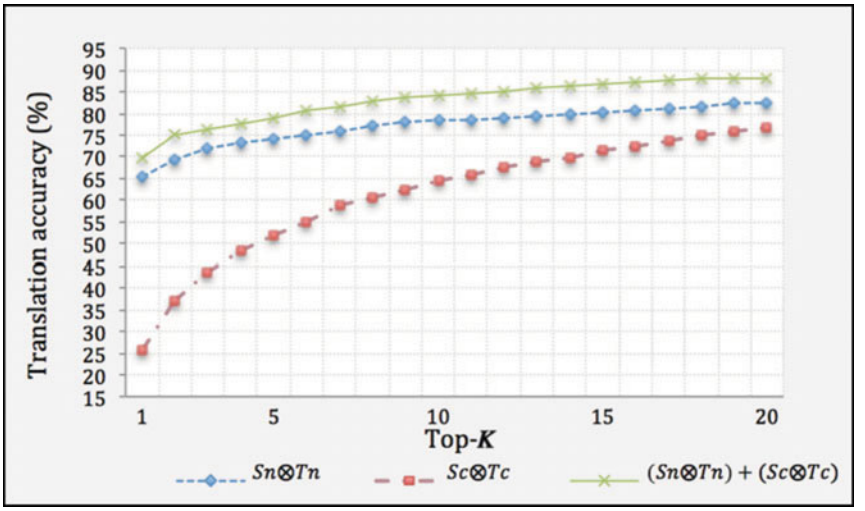


Fig. 4. (Colour online) Top-*k* translation accuracy of the LR trained on the second-order features.

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

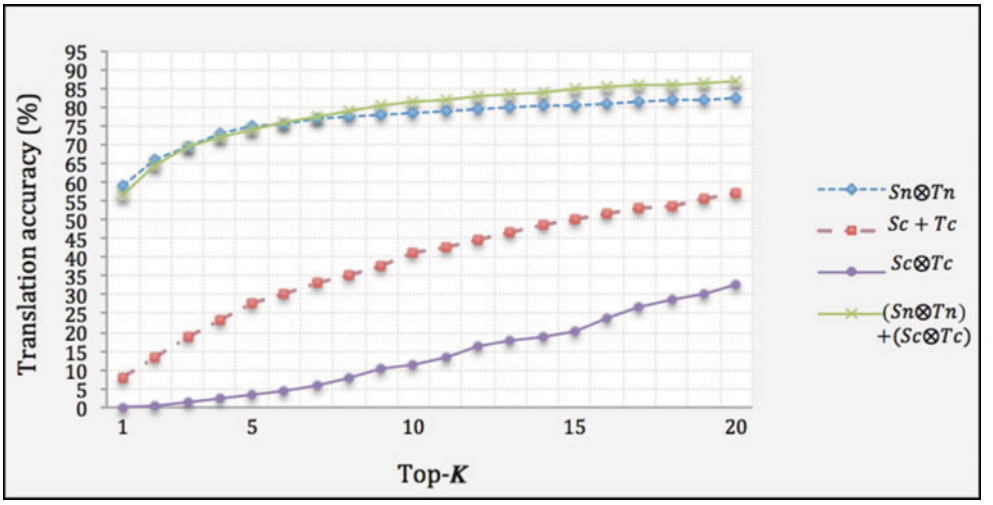


Fig. 5. (Colour online) Top-k translation accuracy of the RF trained on different feature spaces.

to top-8, the translation accuracy of character n-gram and hybrid model do not show significant variation. However, the performance of the hybrid model increased beyond top-8 rank by four per cent compared to the character n-gram model until reaching top-20, where the hybrid model achieved 86.6 per cent and the character n-gram model achieved 82.3 per cent accuracy. Likewise, with the LR, the lowest top-k accuracy is achieved for the context model. As first-order contextual features provide us with better MAP results than the second-order features for the RF, we found it useful to present the top-k translation accuracy for both. Similarly, we obtained higher top-k translation accuracy for first-order contextual features compared to that of second-order features.

5 Conclusion

We propose a method for measuring the similarity between biomedical terms across languages. For this purpose, we use monolingual biomedical terms for both the source and the target languages provided in the form of a bilingual dictionary. The terms are represented using character n-grams and contextual features. Character n-gram models exploit the internal structure of the term, whereas contextual models use the distributional semantics across languages. We further define two types of character n-grams and contextual features, namely first-order and second-order features. We investigate the performance of each feature type separately and in combination through a series of experiments.

Our proposed method differs from prior methods for cross-lingual biomedical term translation detection in that it incorporates two different types of features namely, character n-grams and contextual features, within the same classifier. We evaluate the two feature types using two different classifiers: RF and LR. The experimental results show that the character n-grams are more accurate when predicting the target term translations than the contextual features. Moreover, both the RF and

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

LR classifiers report improved MAP and top-k translation accuracies with the hybrid second-order feature spaces compared to character n-grams and contextual spaces used in isolation. We believe that our method could be of assistance to human annotators when compiling the translation dictionaries for highly specialized domains such as medicine.

There are several aspects of the performance of the proposed method that can be further improved. First, the ambiguity of term translations can be addressed in the model. A single term in the source language can be translated differently into the target language depending on the context in which it is used. Although the proposed method uses contextual features for representing a term, it does not disambiguate the different senses in which a term is used in the target domain. We could extend the proposed method by using a probabilistic model $p(T_m|S_n, c)$, where we could predict the likelihood of translating a source term S_n to the target term T_m given the context c in which S_n has been used.

Second, multiple synonymous words in a source language can be translated into a single term in the target language, and vice versa. For example, *sleeplessness*, *somnolence*, and *somnolency* in English are synonyms and translate to *somnolence* in French. On the other hand, *sore throat* in English is ambiguous and has different translations in French such as *angina*, *pharyngite*, or *mal de gorge* which are not synonyms, because there are slight differences between these terms meaning any of various inflammations of the tonsils, pharynx, or larynx characterized by pain in swallowing. A model that can pre-classify such types of words before they are translated into the target language using the proposed method could potentially solve this issue. We differ these issues to our future work on this topic.

References

Baroni, M., and Lenci, A. 2010. Distributional memory: a general framework for corpus-based semantics. *Computational Linguistics* **36**(4): 673–721.

Bollegala, D., Maehara, T., and ichi Kawarabayashi, K. 2015. Embedding semantic relations into word representations. In *Proceedings of IJCAI*, Buenos Aires, Argentina: AAAI, pp. 1222–8.

Bollegala, D., Matsuo, Y., and Ishizuka, M. 2007. An integrated approach to measuring semantic similarity between words using information available on the web. In *Proceedings of HTL-NAACL'07*, Rochester, NY: ACL, pp. 340–7.

Boström, H. 2007. Estimating class probabilities in random forests. In *International Conference on Machine Learning and Applications*, pp. 211–6.

Breiman, L. 2001. Random forests. *Machine Learning* **45**(1): 5–32.

Chan, Y. S., and Ng, H. T. 2005. Word sense disambiguation with distribution estimation. In *IJCAI'05*, pp. 1010–5.

Chiao, Y.-C., and Zweigenbaum, P. 2002. Looking for candidate translational equivalents in specialized, comparable corpora. In *Proceedings of the 19th International Conference on Computational Linguistics*, Taipei, Taiwan: ACL, pp. 1–5.

Claveau, V. 2008. Automatic translation of biomedical terms by supervised machine learning. In *Proceedings of LREC*, Marrakech, Morocco: European Language Resources Association, pp. 684–91.

Clopper, C. J., and Pearson, E. S. 1934. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* **26**(4): 404–13.

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

Dias, G., Moraliyski, R., Cordeiro, J., Doucet, A., and Ahonen-Myka, H. 2010. Automatic discovery of word semantic relations using paraphrase alignment and distributional lexical semantics analysis. *Natural Language Engineering* **16**(4): 439–67.

Díaz-Uriarte, R., and De Andres, S. A. 2006. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics* **7**(1): 1–13.

Erdmann, M., Nakayama, K., Hara, T., and Nishio, S. 2009. Improving the extraction of bilingual terminology from wikipedia. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* **5**(4): 1–31.

Fan, J.-W., and Friedman, C. 2007. Semantic classification of biomedical concepts using distributional similarity. *Journal of the American Medical Informatics Association* **14**(4): 467–77.

Kontonatsios, G., Korkontzelos, I., Tsujii, J., and Ananiadou, S. 2014a. Combining string and context similarity for bilingual term extraction from comparable corpora. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: ACL, pp. 1701–12.

Kontonatsios, G., Korkontzelos, I., Tsujii, J., and Ananiadou, S. 2014b. Using a random forest classifier to compile bilingual dictionaries of technical terms from comparable corpora. In *Proceedings of the European Chapter for the Association for Computational Linguistics (ACL)*, Gothenburg, Sweden: ACL, pp. 111–6.

Lin, D. 1998. Automatic retrieval and clustering of similar words. In *ACL 1998*, pp. 768–74.

Mcnamee, P., and Mayfield, J. 2004. Character n-gram tokenization for european language text retrieval. *Information Retrieval* **7**(1–2): 73–97.

Mikolov, T., Chen, K., and Dean, J. 2013a. Efficient estimation of word representation in vector space. *CoRR* **abs/1301.3781**.

Mikolov, T., Tau Yih, W., and Zweig, G. 2013b. Linguistic regularities in continous space word representations. In *NAACL'13*, pp. 746–51.

Mitchell, J., and Lapata, M. 2008. Vector-based models of semantic composition. In *ACL-HLT'08*, pp. 236–44.

Nakov, P., and Tiedemann, J. 2012. Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (short-papers)*, Jeju Island, South Korea: ACL, pp. 301–5.

Namer, F., and Baud, R. 2005. Predicting lexical relations between biomedical terms: towards a multilingual morphosemantics-based system. *Studies in Health Technology and Informatics* **116**: 793–8.

Rapp, R. 1999. Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, College Park, Maryland, USA: ACL, pp. 519–26.

Rapp, R. 2008. The automatic generation of thesauri of related words for english, french, german, and russian. *International Journal of Speech Technology* **11**(3–4): 147–56.

Saralegi, X., San Vicente, I., and Gurrutxaga, A. 2008. Automatic extraction of bilingual terms from comparable corpora in a popular science domain. In *Proceedings of Building and using Comparable Corpora Workshop*, Marrakech, Morocco, pp. 27–32.

Tiedemann, J. 2012. Character-based pivot translation for under-resourced languages and domains. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Avignon, France: ACL, pp. 141–51.

Tiedemann, J., and Nakov, P. 2013. Analyzing the use of character-level translation with sparse and noisy datasets. In *Proceedings of Recent Advances in Natural Language Processing*, Hissar, Bulgaria: INCOMA, pp. 676–84.

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press

Turney, P. D., and Pantel, P. 2010. From frequency to meaning: vector space models of semantics. *Journal of Aritificial Intelligence Research* **37**: 141–88.

Vilar, D., Peter, J.-T., and Ney, H. 2007. Can we translate letters?. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic: ACL, pp. 33–9.

Weeds, J., Dowdall, J., Schneider, G., Keller, B., and Weir, D. 2007. Using distributional similarity to organise biomedical terminology. *Application-Driven Terminology Engineering* **2**(97): 107–41.

Xi, N., Tang, G., Dai, X., Huang, S., and Chen, J. 2012. Enhancing statistical machine translation with character alignment. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju, Korea: ACL, 2: 285–90.

https://doi.org/10.1017/S1351324915000431 Published online by Cambridge University Press