2017-9

Idiom Type Identification with Smoothed Lexical Features and a Maximum Margin Classifier

Giancarlo Salton
Technological University Dublin, giancarlo.salton@mydit.ie

Robert J. Ross
Technological University Dublin, robert.ross@tudublin.ie

John D. Kelleher
Technological University Dublin, john.d.kelleher@tudublin.ie

Recommended Citation

Salton, G., Ross, R., Kelleher, J. (2017) Idiom Type Identification with Smoothed Lexical Features and a Maximum Margin Classifier. *International Conference Recent Advances in Natural Language Processing, Bulgaria, 2017*.

Idiom Type Identification with Smoothed Lexical Features and a Maximum Margin Classifier

Giancarlo D. Salton and **Robert J. Ross** and **John D. Kelleher**
School of Computing
Dublin Institute of Technology
Ireland
giancarlo.salton@mydit.ie {robert.ross,john.d.kelleher}@dit.ie

Abstract

In our work we address limitations in the state-of-the-art in idiom type identification. We investigate different approaches for a lexical fixedness metric, a component of the state-of-the-art model. We also show that our Machine Learning based approach to the idiom type identification task achieves an F1-score of 0.85, an improvement of 11 points over the state-of-the-art.

1 Introduction

Idioms are a figurative form of language whose meaning is non-compositional, i.e., their meaning cannot be derived from their individual constituents. Examples of idioms, from English, are *pull one's leg* ("to trick someone by telling something untrue") and *hit the mark* ("be successful in an attempt or accurate in a guess").

Any natural language processing (NLP) system must be able to correctly process and interpret idioms (Villavicencio et al., 2005). A common practice in NLP systems is to use an idiom dictionary as part of the process for handling idioms. However, compiling and maintaining a dictionary by hand is expensive and time consuming. Therefore, reliable ways of automatically identifying idioms are important to keep idiom dictionaries in NLP systems up-to-date (Bannard, 2007).

The automatic construction of idiom dictionaries using idiom type identification (i.e., identifying expressions that have an associated idiom[1]) is an active topic of research. Within this field, researchers have mainly followed two approaches: supervised methods that rely on manually encoded knowledge (e.g., (Copestake et al., 2002) and (Villavicencio et al., 2004)); and unsupervised methods that rely on knowledge extracted from corpora (e.g, (Lin, 1999), (Bannard, 2007) and (Fazly et al., 2009)). Note that to date the supervised methods are idiom-specific and cannot be generalized to a broader class of expressions.

Our research focuses the task of identifying expressions composed of a verb and a noun occurring in its direct object position that have an idiomatic meaning associated with them (Nunberg et al., 1994). These expressions are referred to as VNICs, short for *verb+noun idiomatic combination*. VNICs are the most frequent class of idioms (Villavicencio et al., 2004) and occur across languages (Baldwin and Kim, 2010). We refer to this task as VNIC type identification.

We consider the state-of-the-art method within the field of VNIC type identification to be the work of Fazly et al. (2009). Fazly et al. devise a set of fixedness metrics based on the observation that VNICs are generally more lexically and syntactically fixed than other verb+noun combinations.

In our current work, we identify a number of problems with Fazly et al.'s model and propose modifications to overcome these difficulties. Also, we show that using the fixedness metrics as input features to a Support Vector Machine (SVM) classifier results in an improved method for VNIC type identification compared to Fazly et al.'s model.

The paper is organized as follows: §2 reviews existing fixedness metrics; §3 describes our new fixedness metrics; §4 outlines related work; §5 presents our evaluation methodology; §6 describes the SVM trained using the new fixedness metrics and our results; and §7 presents our conclusions.

2 Fazly et al.'s Fixedness Model

Fazly et al. (2009) present a set of fixedness metrics designed to identify verb+noun pairs that have

[1] As distinct from *idiom token identification* which is the task of distinguishing between idiomatic and literal instances of an expression with an associated idiomatic meaning.

an associated VNIC. Fazly et al. base their approach on the evidence that idioms are more syntactically and lexically fixed than literal constructions. In this approach all verb+noun pairs receive an overall fixedness score that is a linear combination of a syntactic fixedness metric and a lexical fixedness metric. In the following subsections we present and analyse the details of this approach.

2.1 Syntactic Fixedness

Fazly et al. (2009) propose a metric to capture the syntactic fixedness of idioms based on the observation of Riehemann (2001) that idiomatic expressions are expected to appear more frequently under their canonical syntactic form than literal combinations. Fazly et al. describes three types of syntactic variations that can be characteristic of idiomatic combinations: "Passivization"; "Determiner type"; and "Pluralization". Merging these three syntactic variations, they obtained a set $\mathcal{P}$ of eleven syntactic patterns, see Table 1

No.	Verb	Determiner	Noun
pt_1	v_{active}	DET:*NULL*	$n_{singular}$
pt_2	v_{active}	DET:*a/an*	$n_{singular}$
pt_3	v_{active}	DET:*the*	$n_{singular}$
pt_4	v_{active}	DET:*DEM*	$n_{singular}$
pt_5	v_{active}	DET:*POSS*	$n_{singular}$
pt_6	v_{active}	DET:*NULL*	n_{plural}
pt_7	v_{active}	DET:*the*	n_{plural}
pt_8	v_{active}	DET:*DEM*	n_{plural}
pt_9	v_{active}	DET:*POSS*	n_{plural}
pt_{10}	v_{active}	DET:*other*	$n_{singular,plural}$
pt_{11}	$v_{passive}$	DET:*any*	$n_{singular,plural}$

Table 1: Syntactic patterns: the verb *v* can be active (v_{act}) or passive (v_{pass}); the determiner (DET) can be NULL, indefinite (*a/an*), definite (*the*), demonstrative (DEM), or possessive (POSS); the noun *n* can be in singular (n_{sg}) or plural (n_{pl}).

The goal of Fazly et al.'s syntactic fixedness is to compare the behaviour of a target verb+noun pair to the behaviour of a "typical" verb+noun pair. The syntactic behaviour of a "typical" verb+noun pair is defined as a prior distribution over the set $\mathcal{P}$. The syntactic behaviour of the target verb+noun pair is defined as a posterior distribution over the set $\mathcal{P}$ given the pair's constituents. Thus, its syntactic behaviour is calculated as the posterior estimate for a pattern $pt \in \mathcal{P}$[2]. The difference between the behaviour of the target verb+noun pair and the "typical" verb+noun pair is calculated as the divergence between the posterior and the prior distributions over $\mathcal{P}$ as measured using the Kullback-Leibler divergence. Syntactic fixedness takes values in the range $[0, +\infty]$ where larger values denote higher degrees of syntactic fixedness, i.e., the verb+noun pair is more likely to have a VNIC meaning.

2.2 Lexical Fixedness

Typically, idioms do not have lexical variants and, when they have, they are generally unpredictable (Fazly et al., 2009). Therefore, Fazly et al. assume that a verb+noun pair is lexically fixed (and likely to be a VNIC) to the extent that replacing one of its constituents by a semantically similar word does not generate another valid idiomatic combination. Based on this, Fazly et al. propose a measure to capture the degree to which a given verb+noun pair is lexically fixed with respect to the set of its variants. These variants are generated by replacing either the verb or the noun by a word from a set of semantically similar words and is defined as:

$$S_{sim}(v,n) = \{\langle v_i, n\rangle | 1 \leq i \leq K_v\} \cup \{\langle v, n_j\rangle | 1 \leq j \leq K_n\}$$

where $\{\langle v_i, n\rangle | 1 \leq i \leq K_v\}$ is the set of similar combinations generated by replacing the verb by a word from the set of K_v most similar verbs; and $\{\langle v, n_j\rangle | 1 \leq j \leq K_n\}$ is the set of similar combinations generated by replacing the noun by a word from the set of K_n most similar nouns.

To measure the strength of the association between the target verb+noun pair's constituents, Pointwise Mutual Information (PMI) (Church et al., 1991) is applied to the pair and to its set of similar combinations $S_{sim}(v,n)$.

The idea behind lexical fixedness is that the target verb+noun pair $\langle v, n\rangle$ is lexically fixed, and likely to be a VNIC, to the extent its PMI deviates from the mean PMI of the set $S_{sim}(v,n) \cup \langle v, n\rangle$. Following this assumption, lexical fixedness of a verb+noun pair is calculated as a standard z-score:

$$\mathcal{F}_{lex}(v,n) = \frac{PMI(v,n) - \overline{PMI}}{s} \quad (1)$$

[2]For more details on the definition of the prior and posterior distributions see (Fazly et al., 2009).

where $PMI(v,n)$ is the PMI of the target pair $\langle v,n\rangle$; $\overline{PMI}$ and s are the mean and standard deviation of PMI applied to the verb+noun pairs listed in $S_{sim}(v,n) \cup \langle v,n\rangle$. Lexical fixedness falls into the range of $[-\infty,+\infty]$, where higher values mean higher degrees of lexical fixedness (i.e., the verb+noun pair is more likely to be a VNIC).

2.3 Overall Fixedness

Fazly et al. (2009) merge the lexical and syntactic metrics using a weighted linear combination to score the overall fixedness of a verb+noun pair. Note, lexical and syntactic fixedness have different ranges so we rescale them to the range $[0,1]$ before combining them. Overall fixedness is defined as follows:

$$\mathcal{F}_{over}(v,n) = \omega \mathcal{F}_{syn}(v,n) + (1-\omega)\mathcal{F}_{lex}(v,n) \quad (2)$$

where the weight ω controls the relative contribution of each measure for predicting the VNIC's idiomaticity. Values close to 1 means higher degrees of overall fixedness. Moreover, when a particular pair score higher than a certain threshold it is assumed to have an idiomatic expression associated with it, i.e., the pair is identified as a VNIC. In previous work, Fazly et al. used the median value of the test set as the threshold. We see this as problematic and we discuss it in Section 5.2.

2.4 Analysis of Fazly et al.'s Fixedness Model

Fazly et al. have shown their fixedness model to be useful in VNIC type identification. However, their model does have limitations. The definition of lexical fixedness is based on PMI which is known to be biased towards infrequent events (Turney and Pantel, 2010). This property of PMI may lead to undesired results when computing lexical fixedness using counts obtained from a corpus. Furthermore, it is difficult to interpret PMI for those pairs listed in $S_{sim}(v,n)$ that are not observed in the corpus. All pairs are generated by using synonyms (or at least similar related words) and should be acceptable combinations in language. Therefore, the pairs' components should carry information about each other, even if it is small. Thus, we see that just discarding the pairs[3] would affect the result and reduce the power of the model. Therefore, especially for under-resourced languages with small corpora (where the chance of many pairs in $S_{sim}(v,n)$ not being observed is high), a more efficient way to measure the pair's association strength is needed.

[3] In other words, discarding the pairs is the same as setting PMI = 0, following *Information Theory* conventions.

3 Alternative Lexical Fixedness Metrics

In our work we have investigated four ways to replace PMI by other metrics in order to deal with the limitations faced by a lexical fixedness metric based on PMI. In the following we describe these alternatives, and consider how these alternatives can be most effectively combined.

3.1 Probabilities

As mentioned early, a VNIC is expected to be more likely to occur in language than its semantically similar variants. From a probabilistic perspective, we can assume that a VNIC has a higher probability of occurring in language than its semantically similar variants (e.g., literal variants). Following this intuition, we first propose to replace the PMI of a target verb+noun pair by the pair's probability estimated from the corpus as a base for a lexical fixedness metric. The probability for a verb+noun pair is estimated as:

$$P(v,n) = \frac{f(v,n)}{f(*,*)} \quad (3)$$

where $f(v,n)$ is the frequency of the verb+noun pair in a direct object relation (be it a target verb+noun pair or one of its similar variants), occurring in the corpus; and $f(*,*)$ is the frequency of all verb+noun pairs that occur in a direct object relationship in the corpus.

We assume that a target verb+noun pair $\langle v,n\rangle$ is lexically fixed, and likely to be a VNIC, to the extent its probability of occurring in language deviates positively from the mean probability of the set $S_{sim}(v,n) \cup (v,n)$. Thus, we also calculate lexical fixedness based on probabilities as a z-score:

$$\mathcal{F}_{lex} = \frac{P(v,n) - \overline{P}}{s} \quad (4)$$

where $P(v,n)$ is the probability of the target verb+noun pair; and $\overline{P}$ and s are the mean and standard deviation of Equation 3 applied to elements of $S_{sim}(v,n) \cup (v,n)$.

3.2 Smoothed Probabilities

A lexical fixedness metric using raw probabilities may have some of the same disadvantages as a PMI-based metric when estimated with counts

from a corpus. For example, when a verb+noun pair listed in $S_{sim}(v, n)$ does not occur in the corpus we end up with a probability of zero. Just because a particular verb+noun pair does not appear in a corpus this does not mean that this combination cannot occur in language at all.

Inspired by the use of smoothing techniques in language modelling research to overcome the problem of unseen *n*-grams with zero probabilities, for our second metric we cast a VNIC as a *bi*-gram composed of a verb+noun pair ignoring the words in between the verb and the noun. This framing allows us to apply *Modified Kneser-Ney smoothing*[4] to the probabilities of our verb+noun pairs thereby ensuring that all verb+noun pairs in our experiments (including unseen variants listed in $S_{sim}(v, n)$) have non-zero probabilities[5].

As with the lexical fixedness metric based on probabilities, we stick with the same assumption regarding how likely a verb+noun pair is to be a VNIC. Thus, the lexical fixedness based on smoothed probabilities is calculated using a z-score as in Equation (4).

3.3 Interpolated Back-off Probabilities

When estimating a language model from corpora, we may suffer with occurrences of outliers or under-representative samples of *n*-grams (Koehn, 2010). This problem happens if the higher-order *n*-grams are too sparse and, thus, they may be unreliable. The problem is more common when small corpora are used to estimate the model. As we are now considering the verb+noun pair as a *bi*-gram we may also have to face this problem. To overcome these difficulties, it is a common practice to rely on lower-order *n*-grams, which are considered more robust, even if the higher-order *n*-gram have been observed. To do that, one can simply interpolate the high- and low-order *n*-grams into a single probability and, thus, bring together the benefits of longer contexts in higher-order *n*-grams and the robustness of low-order *n*-grams.

A simple but efficient way to interpolate higher- and lower-order *n*-grams is to first apply *Modified Kneser-Ney smoothing* and then sum the smoothed probabilities. We propose to interpolate the probability for the *bi*-gram composed by a verb+noun pair using an interpolation weight of 0.5 the higher- and lower-order *n*-grams (i.e. we give the same weight for *bi*-grams and *one*-grams).

We keep the same assumption of lexical fixedness based on probabilities regarding how likely a verb+noun pair is to be a VNIC. Therefore, the lexical fixedness based on interpolated back-off probabilities is calculated as in Equation 4.

3.4 Normalized Google Distance

So far, we base our propositions on probabilistic and language models approaches. Nevertheless, there are other metrics that can be used to measure the association strength between two words. Thus, we also investigate the use of Normalized Google Distance (NGD) (Cilibrasi and Vitányi, 2007).

NGD is a metric that relies on page counts returned by a search engine on the Internet to measure the strength of the association between two words. As we are interested in VNICs type identification in monolingual corpora, we decided to experiment with an NGD version that uses counts directly extracted from a corpus rather then returned from a search engine on the Internet. A property of NGD that makes it of interest for us to apply is its smooth space of values, granted by removing the dependency of multiplications in the formula. Our NGD variant[6] is defined as:

$$NGD(v, n) = \frac{max\left\{\log(f(v)), \log(f(n))\right\} - \log(f(v, n))}{\log(f(*, *)) - min\left\{\log(f(v)), \log(f(n))\right\}} \quad (5)$$

where $f(v)$ is the frequency of the verb v occurring with any noun as its direct object; $f(n)$ is the frequency of the noun n occurring as a direct object of any transitive verb in the corpus; $f(v, n)$ is the frequency of the verb+noun pair occurring in a direct object relation; and $f(*, *)$ is the frequency of all verb+noun pairs in a direct object relation.

Lexical fixedness based on NGD also follows the assumption that if the NGD of the target verb+noun pair $\langle v, n\rangle$ deviates positively from the mean NGD of the set $S_{sim}(v, n) \cup (v, n)$ then the pair is likely to be a VNIC. Thus, lexical fixedness based on NGD is also calculated as a z-score following Equation 4.

[4]For details on the *Modified Kneser-Ney* please refer to Chen and Goodman (1999).

[5]There are a number of (simpler) smoothing techniques that we could have used, such as add-1 or Laplace smoothing. However, we chose *Modified Kneser-Ney smoothing* as it is considered the state-of-the-art smoothing technique for n-grams, for more see (Brychcín and Konopík, 2014).

[6]Where we also set $\log 0 = 0$, following *Information Theory* conventions, when any of the frequencies involved in NGD calculation is 0.

3.5 Converting the Fixedness Metrics into Classification Models

In the previous section we presented four new lexical fixedness metrics. Similar to Fazly et al., we can use each of these lexical fixedness metrics to compute an overall fixedness score for a verb+noun pair by combining each of them with Fazly et al.'s syntactic fixedness metric using a weighted linear combination as in Equation 2. In the case of our proposed metrics, we replace the original Fazly et al.'s lexical fixedness with one of our own proposed metrics. Including Fazly et al.'s model described in Section 2, these combinations of syntactic and lexical metrics give us the following five models:

1. **Fazly et al.'s model**: Fazly et al.'s syntactic + Fazly et al.'s lexical
2. **Syntactic+Probabilities**: Fazly et al.'s syntactic + Lexical based on probabilities
3. **Syntactic+Smoothed** Fazly et al.'s syntactic + Lexical based on smoothed probabilities
4. **Syntactic+Interpolated**: Fazly et al.'s syntactic + Lexical based on interpolated back-off probabilities
5. **Syntactic+NGD**: Fazly et al.'s syntactic + Lexical based on NGD

While these models provide a real valued measure of the fixedness of a given verb+noun pair, we need to apply a thresholding function to construct a useful classifier. We do this using the logistic function: The logistic function is defined as:

$$sigmoid(x) = \frac{1}{1 + \mathrm{e}^{-(x)}} \quad (6)$$

We applied a different threshold to each of the above models. All the pairs that scored above the threshold for a model were classified as VNICs by that model. The process of setting the threshold for each model is described in Subsection 5.2.

4 Related Work

Although this paper focuses on idiom type identification it is worth noting that there has been a substantial amount of research on idiom token identification. Peng et al. (2014) frame idiom token identification in terms of modeling the global lexical context (essentially using topic models to distinguish idiomatic and literal uses of expressions). Salton et al. (2016) study the use of distributed sentential semantics generated by Sent2Vec (Kiros et al., 2015) to train a general classifier that can take any sentence containing a candidate expression and predict whether the usage of that expression is literal or idiomatic. More recently, Peng and Feldman (2017) presented a model using word embeddings to analyze the context a particular expression is inserted in and predict if its usage is literal or idiomatic.

Turning to the general problem of identifying multiword expressions (MWEs) with associated non-compositional meanings research recent includes Yazdani et al. (2015) model for noun compounds in English and Farahmand and Henderson (2016) work on identifying of collocations. Focusing on the specific task of idiom type identification, Muzny and Zettlemoyer (2013) describes a model that classifies multi-word Wiktionary entries as idiomatic or literal. The model uses a number of lexical and graph based features to calculate the relatedness between the words in the entry and the definition. Muzny and Zettlemoyer (2013) model relies on the Wiktionary structure both in terms of the entry-definition relationship and the definition of the lexical features. Consequently, porting the model for use on an unstructured mono-lingual corpus is non-trivial. Also, although we consider Fazly et al. (2009) the state-of-the-art in VNIC type identification in English, Senaldi et al. (2016) present a model using distributed semantics to identify idiom types in Italian. The authors analysed the differences between idiomatic and literal phrases in embedding spaces, in a similar fashion to lexical fixedness.

Also, previous work has used smoothing and counts obtained from the web for infrequent combinations of words. Keller and Lapata (2003) showed that the web can be used to obtain reliable counts for unseen bigrams but did not evaluate their work on idiom type identification. Ramisch et al. (2010) used the web as a corpus to obtain better counts for n-grams in a language model setting to identify English noun compounds.

5 Evaluating the Classification Models

In order to assess the performance of the classification models we compare them in a classification task over a balanced dataset. We proceed by explaining our data preparation (Subsection 5.1) and describing the methodology to set the thresholds for each model (Subsection 5.2). Finally, we present and discuss the results (Subsection 5.3).

5.1 Data Preparation

We started our data preparation by parsing the written portion of the BNC corpus (Burnard, 2007) using the Stanford Parser (Manning et al., 2014). From the parsed sentences we extracted all verb+noun pairs that occurred in at least one of the syntactic patterns in Table 1. For these extracted verb+noun pairs we recorded the total count of the pair and the total count of the pair in each pattern.

Following the first step, we proceeded by applying Fazly et al.'s syntactic and Fazly et al.'s lexical fixedness metrics and our modified versions of lexical fixedness to all pairs given the recorded counts. To generate the similar combinations required by the lexical fixedness metrics, we used the automatically built thesaurus of Lin (1998). As reported by Fazly et al. (2009), there is little variation on the results for a K number of similar combinations when $20 \leq K \leq 100$, where $K = (K_v + K_n)$. We thus choose $K = 40$ for all lexical models, with $K_v = K_n$[7].

After calculating all fixedness metrics, we kept only those verb+noun combinations which occur at least 10 times in the corpus (note that we did not take into account the determiners introducing the noun). We expect this constraint to balance the distributions of all models tested. Range normalization was then performed on all fixedness metrics before the five overall fixedness scores were determined. To set the weighted linear combination parameter ω, we choose the same value ($\omega = 0.6$) reported by Fazly et al. as the most reasonable choice for all overall models.

Given the five overall scores calculated in the previous step, we selected the top 1,000 verb+noun pairs ranked by each model. Using this methodology, we found only 2,091 different pairs among the 5 lists, i.e., there is an overlap among all metrics. For each of these 2,091 verb+noun pairs, we checked in the Cambridge Idioms[8] and the Collins COBUILD[9] dictionaries whether the verb+noun pair was listed as an idiom[10]. Of the 2,091 pairs, a total of 414 verb+noun pairs were found to be VNICs (and thus, 1,677 were literal combinations). Using this labelled set of 2,091 pairs we created a training and test set.

Fazly et al. used a balanced test set for their evaluations. In order to make our evaluation comparable we also created a balanced test set. To generate our test set we selected VNICs and literal pairs from the 2,091 pairs found in the previous step. From the 414 VNICs, we constrained the selection process so that the selected VNICs occurred with similar frequencies in the corpus as the literal pairs (which were extracted from the 1,677 remaining pairs). This process resulted in a test set of 95 VNICs and 95 literal pairs (called "Test-Set"). The remaining 319 VNICs were held as training data in which we added another 319 verb+noun non-idiomatic pairs, also extracted from the remaining literal pairs (once again with similar frequencies), to create a balanced "Training-Set".

Overall Metric	Threshold
Fazly et al.'s Model	0.63
Syntactic+Probabilities	0.64
Syntactic+Smoothed	0.61
Syntactic+Interpolated	0.62
Syntactic+NGD	0.62

Table 2: Thresholds for each overall metric determined based on the F1-scores on the "Training Set" after applying the logistic function to the scores.

5.2 Setting the Thresholds

In order to create a VNIC type classification model from a fixedness metric we need to apply a threshold to the scores. Fazly et al. choose the median value of their test set as their threshold. We see this as problematic as this provides the model with information about the distribution of the test set and thus biases the evaluation. To avoid this problem, we performed a K-fold cross-validation (with $k = 3$) on our "Training-Set" to find the threshold for each model that maximized the F1-score on the set. This step gave us 5 thresholds, one for each model (see Table 2[11]).

Each model was run on the test set and classified verb+noun pairs with scores greater or equal than the threshold as VNICs and, otherwise, as non-idioms. Table 3 presents the Precision, Recall and

[7] This means we generate 20 similar pairs by changing the verb constituent and 20 similar pairs by changing the noun constituent, with 40 similar combinations in total.

[8] http://dictionary.cambridge.org/

[9] http://www.collinsdictionary.com/

[10] If a verb+noun pair was listed as an idiom in at least one of these two dictionaries we considered it to be a VNIC.

[11] In fact, Fazly et al. recognized the use of the median value as problematic and suggested that a suitable threshold should be determined based on development data.

Model	Pr.	Rec.	F1
Syntactic+Smoothed	**0.83**	**0.78**	**0.77**
Fazly et al.'s Model	0.83	0.75	0.74
Syntactic+Probabilities	0.82	0.73	0.70
Syntactic+Interpolated	0.79	0.64	0.59
Syntactic+NGD	0.78	0.62	0.56

Table 3: Results in terms of precision (Pr.), recall (Rc.) and F1-score (F1) ordered by their F1-scores compared to Standard Fixedness as baseline.

F1-score of the models, ordered by F1-scores.

5.3 Discussion of the Linear Models

Analysing the results in Table 3, we can observe that the worst performance is from the Syntactic+NGD model. We believe the poor results are due to the fact that we are limiting the NGD formulation to consider the counts obtained in the corpus and thus reducing the power of the model. The second worst result is from the Syntactic+Interpolated model, which is slightly higher than the worst model. The intuition for the low results is that, when we apply the interpolation after smoothing the probabilities, we are actually reducing too much the difference between the probability of our target pair and the mean probability of the pair and its variants. In other words, we are over-smoothing the probability distributions across each target pair and its variants.

The Syntactic+Probabilities model, has notable higher scores than the two worst models but it still performs worse than Fazly et al.'s Model. We believe that the difficulties encountered when the similar verb+noun pairs does not occur in the corpus, framed as one of the limitations of the state-of-the-art, are somewhat accentuated when applying raw probabilities as the basis for the lexical fixedness metric. Fazly et al.'s Model scored the second best result over the fixedness models.

The best fixedness model is the Syntactic+Smoothed model. Analysing this model one can also point that the difference between the probability of the target and the mean probability of its variants should be reduced and thus incurring on the same problem as the Syntactic+Interpolated model. Nevertheless, we believe the higher results for this model are due to the fact that when we only smooth the probabilities and not interpolate them, the deviations captured on the z-score are closer to the true deviation. We credit these good results to the use of smoothing to save probability mass for the unseen pairs as it enabled the metric to approximate the true degree of deviation in this lexical fixedness metric. In addition, our results were tested for significance by pairing all the models and applying McNemar's test (McNemar, 1947) to those pairs. We found all $p < 0.05$.

6 Support Vector Machines

An analysis of the scores returned by the fixedness metrics revealed a strong non-linearity in the decision boundary between VNIC and non-idiom verb+noun pairs. One limitation of the VNIC type classifiers in §5 is the weighted linear combination used to merge the syntactic and lexical metrics. This linear approach cannot model non-linear decision boundaries. To overcome this limitation, we trained an SVM classifier (Vapnik, 1995) using the fixedness metrics as inputs.

The SVM is a classification tool designed to find the optimal hyperplane that maximizes the distance between two classes (Zaki and Meira Jr., 2014). An SVM projects the input features into a higher-dimensional feature space and attempts to find a linear separating hyperplane in this higher-dimensional space. The intuition is that a linear separating hyperplane may exist in the higher-dimensional feature space even though the classes are not linearly separable in the original input feature space (Kelleher et al., 2015). For the cases where the classes are not perfectly linearly-separable even in the higher-dimensional feature space the SVM introduces "slack variables" for each datapoint which indicates how much that point violates the separable hyperplane. Then, the goal of the SVM training turns into finding the hyperplane with the maximum margin and that also minimizes the slack terms. This new SVM structure is called a "Soft-margin SVM".

The task of training an SVM with a linear kernel is usually framed as a constrained quadratic programming problem in the dual space. However, in its native form, it is an unconstrained empirical loss minimization including a penalty term for the classifier being learned in direct space (Shalev-Shwartz et al., 2007). Framed this way, SVM can be trained by solving this minimization problem applying Stochastic Gradient Descent (SGD) (Bottou, 2010).

6.1 Building SVM Models from Fixedness Metrics

We used Scikit-Learn (Pedregosa et al., 2011) to train a soft-margin SVM with a linear kernel using SGD training and the fixedness metrics as input features. The training algorithm required two hyper-parameters to be set: an α value (a constant that multiplies the regularization term) and the regularization function. To set these we performed a grid search using k-fold cross-validation (with $k = 3$) over the "Training-Set" using all metrics as features. Based on the results, we set $\alpha = 0.0001$ and the regularization function to be the L2-norm. This step gave us a model which we called "SVM-All". We trained it for 20 epochs.

We also performed feature selection using the weights values of a fitted SVM (Guyon et al., 2002). We selected the three features with the highest weights in "SVM-All": Fazly et al.'s syntactic fixedness, Fazly et al.'s lexical fixedness and the lexical fixedness based on probabilities. Another grid search for the best parameters was performed using k-fold cross-validation (with $k = 3$) over the "Training-Set" using only these three features. Based on the results, we set $\alpha = 0.01$ and set the regularization function to be the L1-norm. This step gave us a model which we called "SVM-Select". We trained this for 20 epochs.

As a matter of comparison, we trained a SVM model using only the original Fazly et al.'s metrics as features. Once again, we performed a grid search using k-fold cross validation (with $k = 3$) over the "Training-Set" to set the α parameter and the regularization function. For this model we set $\alpha = 0.0001$ and the regularization function to be the L2-norm. We called this 2-feature SVM model "SVM-Fazly" and we trained it for 20 epochs.

6.2 Discussion of the SVM Models

Table 4 presents the results of the SVM models, Fazly et al.'s model and the Syntactic+Smoothed (our best linear model). Two SVM models outperformed the fixedness models in the classification task. Surprisingly, the results obtained by the SVM-All model are just slightly higher than those obtained by the fixedness models. We believe the high-dimensional space obtained by this 7-feature SVM is too sparse and thus the classification problem become more difficult. The best general result is from the SVM-Select model. In addition to that, we can observe that the SVM model using only Fazly et al.'s original set of fixedness features had the worst performance in terms of F1-score.

Model	Pr.	Rec.	F1
SVM-Select	**0.87**	**0.85**	**0.85**
SVM-All	0.80	0.78	0.78
Syntactic+Smoothed	0.83	0.78	0.77
Fazly et al.'s Model	0.83	0.75	0.74
SVM-Fazly	0.83	0.73	0.71

Table 4: Precision, Recall and F1-Score results for the 3 SVM models and the 2-best linear models: Fazly et al.'s model Syntactic+Smoothed model.

A final point worth considering is the type of errors each of the models is prone to. Taking the VNIC class as the positive class, most of the errors for the two SVM models and our Syntactic+Smoothed models were false negatives (they classified VNICs as non-VNICS). By comparison, the other models all had higher rates of false positives. In our opinion, for this context, false positives are more problematic than false negatives because a false positive may result in a non-idiom being included in an idiom dictionary. In conclusion, not only do our SVM and Syntactic+Smoothed models outperform the other models in terms of F1 but they are also less prone to false positives. Once again, our results were tested for significance by pairing the models and applying McNemar's test to pairs of models. We found all $p < 0.05$.

7 Conclusions

In this paper we presented four different models to overcome the limitations of the state-of-the-art model for VNIC type identification. We took a probabilistic approach by reinterpreting a previous claim that a VNIC is more likely to appear in language use then its semantically similar variants. In addition, we experimented with a different association measure (Normalized Google Distance) applied to a monolingual corpus.

We have shown that a fixedness model using a lexical metric based on smoothed probabilities out-performs the state-of-the-art model in VNIC type identification. At the same time, we showed that feeding the fixedness metrics to an SVM also improves the F1-score on the same VNIC type identification task by 11 points. We see this work as a significant contribution that will lead to improved models for idiom type identification.

Acknowledgments

This research was partly funded by the ADAPT Centre. The ADAPT Centre is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund. Giancarlo D. Salton would like to thank CAPES ("Coordenação de Aperfeiçoamento de Pessoal de Nível Superior") for his Science Without Borders scholarship, proc n. 9050-13-2.

References

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*, CRC Press, Taylor and Francis Group, Boca Raton, FL.

Colin Bannard. 2007. A measure of syntactic flexibility for automatically identifying multiword expressions in corpora. In *Proceedings of the Workshop on a Broader Perspective on Multiword Expressions (MWE '07)*. pages 1–8.

Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*. pages 177–187.

Tomáš Brychcín and Miloslav Konopík. 2014. Semantic spaces for improving language modeling. *Comput. Speech Lang.* 28(1):192–209.

Lou Burnard. 2007. Reference guide for the british national corpus (xml edition). Technical report, http://www.natcorp.ox.ac.uk/.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language* (13):359–394.

Kenneth Church, William Gale, Patrick Hanks, and Donald Hindle. 1991. Using statistics in lexical analysis. In *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*. Erlbaum, pages 115–164.

Rudi L. Cilibrasi and Paul M.B. Vitányi. 2007. The google similarity distance. *IEEE Transactions On Knowledge And Data Engineering* 19(3):370–383.

Ann Copestake, Fabre Lambeau, Aline Villavicencio, Francis Bond, Timothy Baldwin, Ivan A. Sag, and Dan Flickinger. 2002. Multiword expressions: linguistic precision and reusability. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*. pages 1942–1947.

Meghdad Farahmand and James Henderson. 2016. Modeling the non-substitutability of multiword expressions with distributional semantics and a log-linear model. In *Proceedings of the 12th Workshop on Multiword Expressions, MWE@ACL 2016, Berlin, Germany, August 11, 2016.*.

Afsanesh Fazly, Paul Cook, and Suzanne Stevenson. 2009. Unsupervised type and token identification of idiomatic expressions. In *Computational Linguistics*, volume 35, pages 61–103.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine Learning* 46(1-3):389–422.

John D. Kelleher, Brian Mac Namee, and Aoife D'Arcy. 2015. *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Word Examples and Case Studies*. MIT Press.

Frank Keller and Mirella Lapata. 2003. Using the web to obtain frequencies for unseen bigrams. *Comput. Linguist.* 29(3):459–484. https://doi.org/10.1162/089120103322711604.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28*. pages 3276–3284.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'98)*. pages 768–774.

Dekang Lin. 1999. Automatic identification of non-compositional phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*. pages 317–324.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. pages 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* 12(2):153–157. https://doi.org/10.1007/BF02295996.

Grace Muzny and Luke S. Zettlemoyer. 2013. Automatic idiom identification in wiktionary. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013,*

18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL. pages 1417–1421.

Geoffrey Nunberg, Ivan A. Sag, and Thomas Wasow. 1994. Idioms. *Language* 3(70):491–538.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Jing Peng and Anna Feldman. 2017. *Automatic Idiom Recognition with Word Embeddings*, Springer International Publishing, Cham, pages 17–29.

Jing Peng, Anna Feldman, and Ekaterina Vylomova. 2014. Classifying idiomatic and literal expressions using topic models and intensity of emotions. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 2019–2027.

Carlos Ramisch, Aline Villavicencio, and Christian Boitet. 2010. Web-based and combined language models: A case study on noun compound identification. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '10, pages 1041–1049.

Susanne Riehemann. 2001. *A Constructional Approach to Idioms and Word Formation*. Ph.D. thesis, Stanford University.

Giancarlo D. Salton, Robert J. Ross, and John D. Kelleher. 2016. Idiom token classification using sentential distributed semantics. In *Proceedings of the 54th Annual Meeting on Association for Computational Linguistics*.

Marco Silvio Giuseppe Senaldi, Gianluca E. Lebani, and Alessandro Lenci. 2016. Lexical variability and compositionality: Investigating idiomaticity with distributional semantic models. In *Proceedings of the 12th Workshop on Multiword Expressions, MWE@ACL 2016, Berlin, Germany, August 11, 2016.*.

Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. 2007. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th International Conference on Machine Learning*. pages 807–814.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* (37):141–188.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Aline Villavicencio, Francis Bond, Anna Korhonen, and Diana McCarthy. 2005. Editorial: Introduction to the special issue on multiword expressions: Having a crack at a hard nut. *Comput. Speech Lang.* 19(4):365–377.

Aline Villavicencio, Ann Copestake, Benjamin Waldron, and fabre Lambeau. 2004. Lexical encoding of mwes. In *Proceedings of the Workshop on Multiword Expressions: Integrating Processing (MWE '04)*. pages 80–87.

Majid Yazdani, Meghdad Farahmand, and James Henderson. 2015. Learning semantic composition to detect non-compositionality of multiword expressions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. pages 1733–1742.

Mohammed J. Zaki and Wagner Meira Jr. 2014. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press.