Unsupervised Recognition of Literal and Non-Literal Use of Idiomatic Expressions

Caroline Sporleder and **Linlin Li**
Saarland University
Postfach 15 11 50
66041 Saarbrücken, Germany
{csporled,linlin}@coli.uni-saarland.de

Abstract

We propose an unsupervised method for distinguishing literal and non-literal usages of idiomatic expressions. Our method determines how well a literal interpretation is linked to the overall cohesive structure of the discourse. If strong links can be found, the expression is classified as literal, otherwise as idiomatic. We show that this method can help to tell apart literal and non-literal usages, even for idioms which occur in canonical form.

1 Introduction

Texts frequently contain expressions whose meaning is not strictly literal, such as metaphors or idioms. Non-literal expressions pose a major challenge to natural language processing as they often exhibit lexical and syntactic idiosyncrasies. For example, idioms can violate selectional restrictions (as in *push one's luck* under the assumption that only concrete things can normally be pushed), disobey typical subcategorisation constraints (e.g., *in line* without a determiner before *line*), or change the default assignments of semantic roles to syntactic categories (e.g., in *break sth with X* the argument *X* would typically be an instrument but for the idiom *break the ice* it is more likely to fill a patient role, as in *break the ice with Russia*).

To avoid erroneous analyses, a natural language processing system should recognise if an expression is used non-literally. While there has been a lot of work on recognising idioms (see Section 2), most previous approaches have focused on a *type-based classification*, dividing expressions into "idiom" or "not an idiom" irrespective of their actual use in a discourse context. However, while some expressions, such as *by and large*, always have a non-compositional, idiomatic meaning, many idioms, such as *break the ice* or *spill the beans*, share their linguistic form with perfectly literal expressions (see examples (1) and (2), respectively). For some expressions, such as *drop the ball*, the literal usage can even dominate in some domains. Hence, whether a potentially ambiguous expression has literal or non-literal meaning has to be inferred from the discourse context.

(1) Dad had to break the ice on the chicken troughs so that they could get water.

(2) Somehow I always end up spilling the beans all over the floor and looking foolish when the clerk comes to sweep them up.

Type-based idiom classification thus only addresses part of the problem. While it can automatically compile lists of *potentially* idiomatic expressions, it does not say anything about the idiomaticity of an expression in a particular context. In this paper, we propose a novel, cohesion-based approach for detecting non-literal usages (*token-based idiom classification*). Our approach is unsupervised and similar in spirit to Hirst and St-Onge's (1998) method for detecting malapropisms. Like them, we rely on the presence or absence of cohesive links between the words in a text. However, unlike Hirst and St-Onge we do not require a hand-crafted resource like WordNet or Roget's Thesaurus; our approach is knowledge-lean.

2 Related Work

Most studies on idiom classification focus on type-based classification; few researchers have worked on token-based approaches. Type-based methods frequently exploit the fact that idioms have

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 754–762,
Athens, Greece, 30 March – 3 April 2009. ©2009 Association for Computational Linguistics

a number of properties which differentiate them from other expressions. Apart from not having a (strictly) compositional meaning, they also exhibit some degree of syntactic and lexical fixedness. For example, some idioms do not allow internal modifiers (**shoot the long breeze*) or passivisation (**the bucket was kicked*). They also typically only allow very limited lexical variation (**kick the vessel*, **strike the bucket*).

Many approaches for identifying idioms focus on one of these two aspects. For instance, measures that compute the association strength between the elements of an expression have been employed to determine its degree of compositionality (Lin, 1999; Fazly and Stevenson, 2006) (see also Villavicencio et al. (2007) for an overview and a comparison of different measures). Other approaches use Latent Semantic Analysis (LSA) to determine the similarity between a potential idiom and its components (Baldwin et al., 2003). Low similarity is supposed to indicate low compositionality. Bannard (2007) proposes to identify idiomatic expressions by looking at their syntactic fixedness, i.e., how likely they are to take modifiers or be passivised, and comparing this to what would be expected based on the observed behaviour of the component words. Fazly and Stevenson (2006) combine information about syntactic and lexical fixedness (i.e., estimated degree of compositionality) into one measure.

The few token-based approaches include a study by Katz and Giesbrecht (2006), who devise a supervised method in which they compute the meaning vectors for the literal and non-literal usages of a given expression in the training data. An unseen test instance of the same expression is then labelled by performing a nearest neighbour classification. They report an average accuracy of 72%, though their evaluation is fairly small scale, using only one expression and 67 instances. Birke and Sarkar (2006) model literal vs. non-literal classification as a word sense disambiguation task and use a clustering algorithm which compares test instances to two automatically constructed seed sets (one with literal and one with non-literal expressions), assigning the label of the closest set. While the seed sets are created without immediate human intervention they do rely on manually created resources such as databases of known idioms.

Cook et al. (2007) and Fazly et al. (To appear) propose an alternative method which crucially relies on the concept of *canonical form* (CForm). It is assumed that for each idiom there is a fixed form (or a small set of those) corresponding to the syntactic pattern(s) in which the idiom normally occurs (Riehemann, 2001).[1] The canonical form allows for inflectional variation of the head verb but not for other variations (such as nominal inflection, choice of determiner etc.). It has been observed that if an expression is used idiomatically, it typically occurs in its canonical form. For example, Riehemann (2001, p. 34) found that for decomposable idioms 75% of the occurrences are in canonical form, rising to 97% for non-decomposable idioms.[2] Cook et al. exploit this behaviour and propose an unsupervised method in which an expression is classified as idiomatic if it occurs in canonical form and literal otherwise. Canonical forms are determined automatically using a statistical, frequency-based measure. The authors report an average accuracy of 72% for their classifier.

3 Using Lexical Cohesion to Identify Idiomatic Expressions

3.1 Lexical Cohesion

In this paper we exploit lexical cohesion to detect idiomatic expressions. *Lexical cohesion* is a property exhibited by coherent texts: concepts referred to in individual sentences are typically related to other concepts mentioned elsewhere (Halliday and Hasan, 1976). Such sequences of semantically related concepts are called *lexical chains*. Given a suitable measure of semantic relatedness, such chains can be computed automatically and have been used successfully in a number of NLP applications, starting with Hirst and St-Onge's (1998) seminal work on detecting real-word spelling errors. Their approach is based on the insight that misspelled words do not "fit" their context, i.e., they do not normally participate in lexical chains. Content words which do not belong to any lexical chain but which are orthographically close to words which do, are therefore good candidates for spelling errors.

Idioms behave similarly to spelling errors in that they typically also do not exhibit a high de-

[1] This is also the form in which an idiom is usually listed in a dictionary.

[2] Decomposable idioms are expressions such as *spill the beans* which have a composite meaning whose parts can be mapped to the words of the expression (e.g., *spill*→'reveal', *beans*→'secret').

gree of lexical cohesion with their context, at least not if one assumes a *literal meaning* for their component words. Hence if the component words of a potentially idiomatic expression do not participate in any lexical chain, it is likely that the expression is indeed used idiomatically, otherwise it is probably used literally. For instance, in example (3), where the expression *play with fire* is used in a literal sense, the word *fire* does participate in a chain (shown in bold face) that also includes the words *grilling*, *dry-heat*, *cooking*, and *coals*, while for the non-literal usage in example (4) there are no chains which include *fire*.[3]

(3) **Grilling** outdoors is much more than just another **dry-heat cooking** method. It's the chance to play with **fire**, satisfying a primal urge to stir around in **coals** .

(4) And PLO chairman Yasser Arafat has accused Israel of playing with fire by supporting HAMAS in its infancy.

Unfortunately, there are also a few cases in which a cohesion-based approach fails. Sometimes an expression is used literally but does not feature prominently enough in the discourse to participate in a chain, as in example (5) where the main focus of the discourse is on the use of morphine and not on children playing with fire.[4] The opposite case also exists: sometimes idiomatic usages do exhibit lexical cohesion on the component word level. This situation is often a consequence of a deliberate "play with words", e.g. the use of several related idioms or metaphors (see example (6)). However, we found that both cases are relatively rare. For instance, in a study of 75 literal usages of various expressions, we only discovered seven instances in which no relevant chain could be found, including some cases where the context was too short to establish the cohesive structure (e.g., because the expression occurred in a headline).

(5) Chinamasa compared McGown's attitude to morphine to a child's attitude to playing with fire – a lack of concern over the risks involved.

(6) Saying that the Americans were "playing with **fire**" the official press speculated that the "**gunpowder** barrel" which is Taiwan might well "**explode**" if Washington and Taipei do not put a stop to their "**incendiary** gesticulations."

3.2 Modelling Semantic Relatedness

While a cohesion-based approach to token-based idiom classification should be intuitively successful, its practical usefulness depends crucially on the availability of a suitable method for computing semantic relatedness. This is currently an area of active research. There are two main approaches. Methods based on manually built lexical knowledge bases, such as WordNet, model semantic relatedness by computing the shortest path between two concepts in the knowledge base and/or by looking at word overlap in the glosses (see Budanitsky and Hirst (2006) for an overview). Distributional approaches, on the other hand, rely on text corpora, and model relatedness by comparing the contexts in which two words occur, assuming that related words occur in similar context (e.g., Hindle (1990), Lin (1998), Mohammad and Hirst (2006)). More recently, there has also been research on using Wikipedia and related resources for modelling semantic relatedness (Ponzetto and Strube, 2007; Zesch et al., 2008).

All approaches have advantages and disadvantages. WordNet-based approaches, for instance, typically have a low coverage and only work for so-called "classical relations" like hypernymy, antonymy etc. Distributional approaches usually conflate different word senses and may therefore lead to unintuitive results. For our task, we need to model a wide range of semantic relations (Morris and Hirst, 2004), for example, relations based on some kind of functional or situational association, as between *fire* and *coal* in (3) or between *ice* and *water* in example (1). Likewise we also need to model relations between non-nouns, for instance between *spill* and *sweep up* in example (2). Some relations also require world-knowledge, as in example (7), where the literal usage of *drop the ball* is not only indicated by the presence of *goalkeeper* but also by knowing that Wayne Rooney and Kevin Campbell are both football players.

(7) When **Rooney** collided with the **goalkeeper**, causing him to drop the **ball**, **Kevin Campbell** followed in.

We thus decided against a WordNet-based measure of semantic relatedness, opting instead for a distributional approach, *Normalized Google Distance* (NGD, see Cilibrasi and Vitanyi (2007)), which computes relatedness on the basis of page counts returned by a search engine. NGD is a measure of association that quantifies the strength of a

[3] Idioms may, of course, link to the surrounding discourse with their *idiomatic meaning*, i.e., for *play with fire* one may expect other words in the discourse which are related to the concept "danger".

[4] Though one could argue that there is a chain linking *child* and *play* which points to the literal usage here.

relationship between two words. It is defined as follows:

$$NGD(x,y) = \frac{max\{log\ f(x), log\ f(y)\} - log\ f(x,y)}{log\ M - min\{log\ f(x), log\ f(y)\}} \tag{8}$$

where x and y are the two words whose association strength is computed (e.g., *fire* and *coal*), $f(x)$ is the page count returned by the search engine for the term x (and likewise for $f(y)$ and y), $f(x,y)$ is the page count returned when querying for “x AND y” (i.e., the number of pages that contain both, x and y), and M is the number of web pages indexed by the search engine. The basic idea is that the more often two terms occur together relative to their overall occurrence the more closely they are related. For most pairs of search terms the NGD falls between 0 and 1, though in a small number of cases NGD can exceed 1 (see Cilibrasi and Vitanyi (2007) for a detailed discussion of the mathematical properties of NGD).

Using web counts rather than bi-gram counts from a corpus as the basis for computing semantic relatedness was motivated by the fact that the web is a significantly larger database than any compiled corpus, which makes it much more likely that we can find information about the concepts we are looking for (thus alleviating data sparseness). The information is also more up-to-date, which is important for modelling the kind of world knowledge about named entities we need to resolve examples like (7). Furthermore, it has been shown that web counts can be used as reliable proxies for corpus-based counts and often lead to better statistical models (Zhu and Rosenfeld, 2001; Lapata and Keller, 2005).

To obtain the web counts we used Yahoo rather than Google because we found Yahoo gave us more stable counts over time. Both the Yahoo and the Google API seemed to have problems with very high frequency words, so we excluded those cases. Effectively, this amounted to filtering out function words. As it is difficult to obtain reliable figures for the number of pages indexed by a search engine, we approximated this number (M in formula (8) above) by setting it to the number of hits obtained for the word *the*, assuming that this word occurs in virtually all English language pages (Lapata and Keller, 2005). When generating the queries we made sure that we queried for all combinations of inflected forms (for example “fire AND coal” would be expanded to “fire AND coal”, “fires AND coal”, “fire AND coals”, and “fires AND coals”). The inflected forms were generated by the *morph* tools developed at the University of Sussex (Minnen et al., 2001).[5]

3.3 Cohesion-based Classifiers

We implemented two cohesion-based classifiers: the first one computes the **lexical chains** for the input text and classifies an expression as literal or non-literal depending on whether its component words participate in any of the chains, the second classifier builds a **cohesion graph** and determines how this graph changes when the expression is inserted or left out.

Chain-based classifier Various methods for building lexical chains have been proposed in the literature (Hirst and St-Onge, 1998; Barzilay and Elhadad, 1997; Silber and McCoy, 2002) but the basic idea is as follows: the content words of the text are considered in sequence and for each word it is determined whether it is similar enough to (the words in) one of the existing chains to be placed in that chain, if not it is placed in a chain of its own. Depending on the chain building algorithm used, a word is placed in a chain if it is related to *one* other word in the chain or to *all* of them. The latter strategy is more conservative and tends to lead to shorter but more reliable chains and it is the method we adopted here.[6] Note that the chaining algorithm has a free parameter, namely a threshold which has to be surpassed to consider two words related (*relatedness threshold*).

On the basis of the computed chains, the classifier has to decide whether the target expression is used literally or not. A simple strategy would classify an expression as literal whenever one or more of its component words participates in *any* chain. However, as the chains are potentially noisy, this may not be the best strategy. We therefore also evaluate the strength of the chain(s) in which the expression participates. If a component word of the expression participates in a long chain (and is related to all words in the chain, as we require)

[5] The tools are available at: `http://www.informatics.susx.ac.uk/research/groups/nlp/carroll/morph.html`.

[6] If a WordNet-based relatedness measure is used, the chaining algorithm has to perform word sense disambiguation as well. As we use a distributional relatedness measure which conflates different senses anyway, we do not have to disambiguate here.

then this is good evidence that the expression is indeed used in a literal sense. For instance, in (3) the word *fire* belongs to the relatively long chain *grilling – dry-heat – cooking – fire – coals*, providing strong evidence of literal usage of *play with fire*. To determine the strength of the evidence in favour of a literal interpretation, we take the longest chain in which any of the component words of the idiom participate[7] and check whether this is above a predefined threshold (the *classification threshold*). Both the relatedness threshold and the classification threshold are set empirically by optimising on a manually annotated development set (see Section 4.2).

Graph-based classifier The chain-based classifier has two parameters which need to be optimised on labelled data, making this method weakly supervised. To overcome this drawback, we designed a second classifier which does not have free parameters and is thus fully unsupervised. This classifier relies on *cohesion graphs*. The vertices of such a cohesion graph correspond to the (content) word tokens in the text, each pair of vertices is connected by an edge and the edges are weighted by the semantic relatedness (i.e., the inverse NGD) between the two words. The cohesion graph for example (1) is shown in Figure 1 (for expository reasons, edge weights are excluded from the figure). Once we have built the cohesion graph we compute its connectivity (defined as the average edge weight) and compare it to the connectivity of the graph that results from removing the (component words of the) target expression. For instance in Figure 1, we would compare the connectivity of the graph as it is shown to the connectivity that results from removing the dashed edges. If removing the idiom words from the graph leads to a higher connectivity, we assume that the idiom is used non-literally, otherwise we assume it is used literally. In Figure 1, for example, most edges would have a relatively low weight, indicating a weak relation between the words they link. The edge between *ice* and *water*, however, would have a higher weight. Removing *ice* from the graph would therefore lead to a decreased connectivity and the classifier would predict that *break the ice* is used in the literal sense in example (1). Effectively, we replace the explicit thresholds of the lexical chain method by an *implicit threshold* (i.e., change in connectivity), which does not have to be optimised.

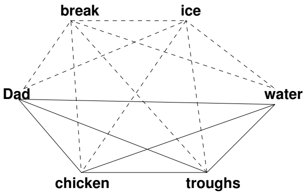

Figure 1: Cohesion graph for example (1)

4 Evaluating the Cohesion-Based Approach

We tested our two cohesion-based classifiers as well as a supervised classifier on a manually annotated data set. Section 4.2 gives details of the experiments and results. We start, however, by describing the data used in the experiments.

4.1 Data

We chose 17 idioms from the *Oxford Dictionary of Idiomatic English* (Cowie et al., 1997) and other idiom lists found on the internet. The idioms were more or less selected randomly, subject to two constraints: First, because the focus of the present study is on distinguishing literal and non-literal usage, we chose expressions for which we assumed that the literal meaning was not too infrequent. We thus disregarded expressions like *play the second fiddle* or *sail under false colours*. Second, in line with many previous approaches to idiom classification (Fazly et al., To appear; Cook et al., 2007; Katz and Giesbrecht, 2006), we focused mainly on expressions of the form V+NP or V+PP as this is a fairly large group and many of these expressions can be used literally as well, making them an ideal test set for our purpose. However, our approach also works for expressions which match a different syntactic pattern and to test the generality of our method we included a couple of these in the data set (e.g., *get one's feet wet*). For the same reason, we also included some expressions for which we could not find a literal use in the corpus (e.g., *back the wrong horse*).

For each of the 17 expressions shown in Table 1, we extracted all occurrences found in the Gigaword corpus that were in canonical form (the forms listed in the table plus inflectional varia-

[7] Note, that it is not only the noun that can participate in a chain. In example (2), the word *spill* can be linked to *sweep up* to provide evidence of literal usage.

tions of the head verb).[8] Hence, for *rock the boat* we would extract *rocked the boat* and *rocking the boat* but not *rock a boat*, *rock the boats* or *rock the ship*. The motivation for this was two-fold. First, as was discussed in Section 2, the vast majority of idiomatic usages are in canonical form. This is especially true for non-decomposable idioms (most of our 17 idioms), where only around 3% of the idiomatic usages are not in canonical form. Second, we wanted to test whether our approach would be able to detect literal usages in the set of canonical form expressions as this is precisely the set of expressions that would be classified as idiomatic by the unsupervised CForm classifier (Cook et al. (2007), Fazly et al. (To appear)). While expressions in the canonical form are more likely to be used idiomatically, it is still possible to find literal usages as in examples (1) and (2). For some expressions, such as *drop the ball* the literal usage even outweighs the non-literal usage. These literal usages would be mis-classified by the CForm classifier.

In principle, though, our approach is very general and would also work on expressions that are not in canonical form and expressions whose idiomatic status is unclear, i.e., we do not necessarily require a predefined set of idioms but could run the classifiers on any V+NP or V+PP chunk.

For each extracted example, we included five paragraphs of context (the current paragraph plus the two preceding and following ones).[9] This was the context used by the classifiers. The examples were then labelled as “literal” or “non-literal” by an experienced annotator. If the distinction could not be made reliably, e.g., because the context was not long enough to disambiguate, the annotator was allowed to annotate “?”. These cases were excluded from the data sets. To estimate the reliability of our annotation, a randomly selected sample (300 instances) was annotated independently by a second annotator. The annotations deviated in eight cases from the original, amounting to an inter-annotator agreement of over 97% and a kappa score of 0.7 (Cohen, 1960). All deviations were cases in which one of the annotators chose “?”, often because there was not sufficient context and the annotation decision had to be made on the basis of world knowledge.

expression	literal	non-literal	all
back the wrong horse	0	25	25
bite off more than one can chew	2	142	144
bite one’s tongue	16	150	166
blow one’s own trumpet	0	9	9
bounce off the wall*	39	7	46
break the ice	20	521	541
drop the ball*	688	215	903
get one’s feet wet	17	140	157
pass the buck	7	255	262
play with fire	34	532	566
pull the trigger*	11	4	15
rock the boat	8	470	478
set in stone	9	272	281
spill the beans	3	172	175
sweep under the carpet	0	9	9
swim against the tide	1	125	126
tear one’s hair out	7	54	61
all	862	3102	3964

Table 1: Idiom statistics (* indicates expressions for which the literal usage is more common than the non-literal one)

4.2 Experimental Set-Up and Results

For the lexical chain classifier we ran two experiments. In the first, we used the data for one expression (*break the ice*) as a development set for optimising the two parameters (the relatedness threshold and the classification threshold). To find good thresholds, a simple hill-climbing search was implemented during which we increased the relatedness threshold in steps of 0.02 and the classification threshold (governing the minimum chain length needed) in steps of 1. We optimised the F-Score for the literal class, though we found that the selected parameters varied only minimally when optimising for accuracy. We then used the parameter values determined in this way and applied the classifier to the remainder of the data.

The results obtained in this way depend to some extent on the data set used for the parameter setting.[10] To control this factor, we also ran another experiment in which we used an oracle to set the parameters (i.e., the parameters were optimised for the complete set). While this is not a realistic scenario as it assumes that the labels of the test data are known during parameter setting, it does provide an *upper bound* for the lexical chain method.

For comparison, we also implemented an **informed baseline classifier**, which employs a simple model of cohesion, classifying expressions as

[8] The extraction was done via manually built regular expressions.

[9] Note that paragraphs tend to be rather short in newswire. For other genres it may be sufficient to extract one paragraph.

[10] We also ran the experiment for different development sets and found that there was a relatively high degree of variation in the parameters selected and in the results obtained with those settings.

literal if the noun inside the expression (e.g., *ice* for *break the ice*) is repeated elsewhere in the context, and non-literal otherwise. One would expect this classifier to have a high precision for literal expressions but a low recall.

Finally, we implemented a **supervised classifier**. Supervised classifiers have been used before for this task, notably by Katz and Giesbrecht (2006). Our approach is slightly different: instead of creating meaning vectors we look at the word overlap[11] of a test instance with the literal and non-literal instances in the training set (for the same expression) and then assign the label of the closest set.

That such an approach might be promising becomes clear when one looks at some examples of literal and non-literal usage. For instance, non-literal examples of *break the ice* occur frequently with words such as *diplomacy*, *relations*, *dialogue* etc. Effectively these words form lexical chains with the *idiomatic* meaning of *break the ice*. They are absent for literal usages. A supervised classifier can learn which terms are indicative of which usage. Note that this information is expression-specific, i.e., it is not possible to train a classifier for *play with fire* on labelled examples for *break the ice*. This makes the supervised approach quite expensive in terms of annotation effort as data has to be labelled for each expression. Nonetheless, it is instructive to see how well one could do with this approach. In the experiments, we ran the supervised classifier in leave-one-out mode on each expression for which we had literal examples.

Table 2 shows the results for the five classifiers discussed above: the informed baseline classifier (Rep), the cohesion graph (Graph), the lexical chain classifier with the parameters optimised on *break the ice* (LC), the lexical chain classifier with the parameters set by an oracle (LC-O), and the supervised classifier (Super). The table also shows the accuracy that would be obtained by a CForm classifier (Cook et al., 2007; Fazly et al., To appear) with gold standard canonical forms. This classifier would label all examples in our data set as "non-literal" (it is thus equivalent to a majority class baseline). Since the majority of examples is indeed used idiomatically, this classifier achieves a relatively high accuracy. However, accuracy is not the best evaluation measure here because we are interested in detecting literal usages among the canonical forms. Therefore, we also computed the precision (P_l), recall (R_l), and F-score (F_l) for the literal class.

	CForm	Rep	Graph	LC	LC-O	Super
Acc	78.25	79.06	79.61	80.50	80.42	95.69
P_l	-	70.00	52.21	62.26	53.89	84.62
R_l	-	5.96	67.87	26.21	69.03	96.45
F_l	-	10.98	59.02	36.90	60.53	90.15

Table 2: Accuracy, literal precision (P_l), recall (R_l), and F-Score (F_l) for the classifiers

It can be seen that all classifiers obtain a relatively high accuracy but vary in precision, recall and F-Score. For the CForm classifier, precision, recall, and F-Score are undefined as it does not label any examples as "literal". As expected the baseline classifier, which looks for repetitions of the component words of the target expression, has a relatively high precision, showing that the expression is typically used in the literal sense if part of it is repeated in the context. The recall, though, is very low, indicating that lexical repetition is not a sufficient signal for literal usage.

The graph-based classifier and the globally optimised lexical chain classifier (LC-O) outperform the other two unsupervised classifiers (CForm and Rep), with an F-Score of around 60%. For both classifiers recall is higher than precision. Note, however, that this is an upper bound for the lexical chain classifier that would not be obtained in a realistic scenario. An example of the values that can be expected in a realistic setting (with parameter optimisation on a development set that is separate from the test set) is shown in column five (LC). Here the F-Score is much lower due to lower recall. This classifier is too conservative when creating the chains and deciding how to interpret the chain structure; it thus only rarely outputs the literal class. The reason for this conservatism may be that literal usages of *break the ice* (the development data) tend to have very strong chains, hence when optimising the parameters for this data set, it pays to be conservative. It is positive to note that the (unsupervised) graph-based classifier performs just as well as the (weakly supervised) chain-based classifier does under optimal circumstances. This means that one can by-pass the parameter setting and the need to label development data by employing the graph-based method.

Finally, as expected, the supervised classifier

[11] We used the Dice coefficient as implemented in Ted Pedersen's Text::Similarity module: `http://www.d.umn.edu/~tpederse/text-similarity.html`.

outperforms all other classifiers. It does so by a large margin, which is surprising given that it is based on relatively simplistic model. This shows that the context in which an expression occurs can really provide vital cues about its idiomaticity. Note that our results are noticeably higher than those reported by Cook et al. (2007), Fazly et al. (To appear) and Katz and Giesbrecht (2006) for similar supervised classifiers. We believe that this may be partly explained by the size of our data set which is significantly larger than the ones used in these studies.

To assess how well our cohesion-based approach works for different idioms, we also computed the accuracy of the graph-based classifier for each expression individually (Table 3). We report accuracy here rather than literal F-Score as the latter is often undefined for the individual data sets (either because all examples of an expression are non-literal or because the classifier only predicts non-literal usages). It can be seen that the performance of the classifier is generally relatively stable, with accuracies above 50% for most idioms.[12] In particular, the classifier performs well on both, expressions with a dominant non-literal meaning and those with a dominant literal meaning; it is not biased towards the non-literal class. For expressions with a dominant literal meaning like *drop the ball*, it correctly classifies more items as “literal” (530 items, 472 of which are correct) than as “non-literal” (373 items, 157 correct).

expression	Accuracy
back the wrong horse	68.00
bite off more than one can chew	79.17
bite one’s tongue	37.35
blow one’s own trumpet	11.11
bounce off the wall*	47.82
break the ice	85.03
drop the ball*	69.66
get one’s feet wet	64.33
pass the buck	82.44
play with fire	82.33
pull the trigger*	60.00
rock the boat	98.95
set in stone	85.41
spill the beans	83.43
sweep under the carpet	88.89
swim against the tide	93.65
tear one’s hair out	49.18

Table 3: Accuracies of the graph-based classifier on each of the expressions (* indicates a dominant literal usage)

5 Conclusion

In this paper, we described a novel method for token-based idiom classification. Our approach is based on the observation that literally used expressions typically exhibit cohesive ties with the surrounding discourse, while idiomatic expressions do not. Hence idiomatic expressions can be detected by the absence of such ties. We propose two methods that exploit this behaviour, one based on lexical chains, the other based on cohesion graphs.

We showed that a cohesion-based approach is well suited for distinguishing literal and non-literal usages, even for expressions in canonical form which tend to be largely idiomatic and would all be classified as non-literal by the previously proposed CForm classifier. Moreover, our findings suggest that the graph-based method performs nearly as well as the best performance to be expected for the chain-based method. This means that the task can be addressed in a completely unsupervised way.

While our results are encouraging they are still below the results obtained by a basic supervised classifier. In future work we would like to explore whether better performance can be achieved by adopting a bootstrapping strategy, in which we use the examples about which the unsupervised classifier is most confident (i.e., those with the largest difference in connectivity in either direction) as input for a second stage supervised classifier.

Another potential improvement has to do with the way in which the cohesion graph is computed. Currently the graph includes all content words in the context. This means that the graph is relatively big and removing the potential idiom often does not have a big effect on the connectivity; all changes in connectivity are fairly close to zero. In future, we want to explore intelligent strategies for pruning the graph (e.g., by including a smaller context). We believe that this might result in more reliable classifications.

Acknowledgments

This work was funded by the German Research Foundation DFG (under grant PI 154/9-3 and the MMCI Cluster of Excellence). Thanks to Anna Mündelein for her help with preparing the data and to Marco Pennacchiotti and Josef Ruppenhofer, for feedback and comments.

[12] Note that the data set for the worst performing idiom, *blow one’s own trumpet* only contained 9 instances. Hence, the low performance for this idiom may well be accidental.

References

Timothy Baldwin, Colin Bannard, Takaaki Tanaka, and Dominic Widdows. 2003. An empirical model of multiword expression decomposability. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment*, pages 89–96.

Colin Bannard. 2007. A measure of syntactic flexibility for automatically identifying multiword expressions in corpora. In *Proceedings of the ACL-07 Workshop on A Broader Perspective on Multiword Expressions*, pages 1–8.

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL-97 Intelligent Scalable Text Summarization Workshop (ISTS-97)*.

Julia Birke and Anoop Sarkar. 2006. A clustering approach for the nearly unsupervised recognition of nonliteral language. In *Proceedings of EACL-06*, pages 329–336.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of semantic distance. *Computational Linguistics*, 32(1):13–47.

Rudi L. Cilibrasi and Paul M.B. Vitanyi. 2007. The Google similarity distance. *IEEE Trans. Knowledge and Data Engineering*, 19(3):370–383.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurements*, 20:37–46.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2007. Pulling their weight: Exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In *Proceedings of the ACL-07 Workshop on A Broader Perspective on Multiword Expressions*, pages 41–48.

A.P. Cowie, R. Mackin, and I.R. McCaig. 1997. *Oxford dictionary of English idioms*. Oxford University Press.

Afsaneh Fazly and Suzanne Stevenson. 2006. Automatically constructing a lexicon of verb phrase idiomatic combinations. In *Proceedings of EACL-06*.

Afsaneh Fazly, Paul Cook, and Suzanne Stevenson. To appear. Unsupervised type and token identification of idiomatic expressions. *Computational Linguistics*.

M.A.K. Halliday and R. Hasan. 1976. *Cohesion in English*. Longman House, New York.

Donald Hindle. 1990. Noun classification from predicate-argument structures. In *Proceedings of ACL-90*, pages 268–275.

Graeme Hirst and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An electronic lexical database*, pages 305–332. The MIT Press.

Graham Katz and Eugenie Giesbrecht. 2006. Automatic identification of non-compositional multiword expressions using latent semantic analysis. In *Proceedings of the ACL/COLING-06 Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 12–19.

Mirella Lapata and Frank Keller. 2005. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing*, 2:1–31.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of ACL-98*, pages 768–774.

Dekang Lin. 1999. Automatic identification of non-compositional phrases. In *Proceedings of ACL-99*, pages 317–324.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Saif Mohammad and Graeme Hirst. 2006. Distributional measures of concept-distance: A task-oriented evaluation. In *Proceedings of EMNLP-06*.

Jane Morris and Graeme Hirst. 2004. Non-classical lexical semantic relations. In *HLT-NAACL-04 Workshop on Computational Lexical Semantics*, pages 46–51.

Simone Paolo Ponzetto and Michael Strube. 2007. Knowledge derived from Wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research*, 30:181–212.

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

H. Gregory Silber and Kathleen F. McCoy. 2002. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28(4):487–496.

Aline Villavicencio, Valia Kordoni, Yi Zhang, Marco Idiart, and Carlos Ramisch. 2007. Validation and evaluation of automatically acquired multiword expressions for grammar engineering. In *Proceedings of EMNLP-07*, pages 1034–1043.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Using wiktionary for computing semantic relatedness. In *Proceedings of AAAI-08*, pages 861–867.

Xiaojin Zhu and Ronald Rosenfeld. 2001. Improving trigram language modeling with the world wide web. In *Proceedings of ICASSP-01*.