Miriam Connor
CS224N
December 6, 2012

CS224N Final Project
Figurative Language Classification of English Manner of Motion Verbs
for a Machine Translation System

Abstract

Being able to differentiate between figurative and literal words and phrases is useful for a variety of natural language processing tasks, including machine translation. In this project, I implemented a linear classifier with a logistic loss function to distinguish between figurative and literal uses of English manner of motion verbs. The feature inputs to this models were the lemma of the manner of motion verb, the types of syntactic dependents attached to it, and the semantic hypernyms of the verb's syntactic dependents, extracted with WordNet. The linear classifier achieved 86% accuracy on a set of test sentences, compared to 77.5% baseline accuracy of a bag-of-words classifier. Finally, I test the effect of adding figurative/literal annotations to English manner of motion verbs on the performance of the Moses machine translation system. I find an increase in BLEU score for annotated inputs, but discuss issues with the experiment that make this result difficult to interpret.

Note: Some of the work for this project was done in conjunction with my final project for CS221: Artificial Intelligence: Principles and Techniques. The work for CS221 mainly comprises the implementation and analysis of a linear classifier model. The work for CS224N comprises the feature design and extraction, the in-depth error analysis of the classifier's performance, and the experiment with the Moses machine translation system.

1. Introduction

While much information exists on the systematic differences between languages, current statistical machine translation systems tend include very little of this information in rendering translations. For example, in Germanic languages like English, verbs that describe a manner of motion, like *walk*, *stumble*, *trot*, or *lope*, are common and widely used, especially in figurative language (e.g., *we walked out of negotiations*, *he stumbled over his words*). However, Romance languages like French have much smaller vocabularies of these verbs, place many more semantic and syntactic restrictions on their use, and make use of their figurative sense much more rarely. This language difference leads to mismatches in the parallel corpora, such as in the following excerpts from Europarl, a parallel corpus containing the proceedings of the European Parliament in 21 languages including English and French.

English: If we allow the postal services to **drift** into complete privatization then we will lose all that…
French: Si nous autorisons la privatisation complète des services postaux, nous perdrons tout cela…
Gloss: *If we authorize the complete privatization of the postal services, we will lose all that…*

The French uses a different phrase than the English to describe privatization of postal services, completely omitting figurative expression with the manner of motion verb *drift*. Mismatches like the above lead to incorrect word alignments and thus and poor performance on manner verb translation from English to French by machine translation systems.

The translation of these verbs from English to French may be improved if the English verbs are simply labeled as *literal* or *figurative*. A variety approaches have been taken for performing literal/figurative distinctions on English expressions—see Shutova (2011) for a review of computational approaches to identifying figurative language. These approaches include mining lexical resources (Peters and Peters 2000), clustering approaches (Birke and Sankar 2006), and inference of mappings between different document domains (Mason 2004). The approach I take is inspired by Gedigan et al (2006). They use a maximum entropy classifier with syntactic and semantic features from FrameNet and PropBank to determine if sentences involving spacial motion and health in the Wall Street Journal corpus are figurative or literal. Their approach takes advantage of a hand-parsed corpus with ProbBank semantic annotations, and they achieve an accuracy of only 95.12%, compared to the 92.90% baseline of assigning all verbs to *figurative*. I attempt to improve on their results using a data set with no annotation other than the figurative/literal class of the verb.

In this report, I describe the design and performance of a linear classifier system that distinguishes between literal and figurative uses of manner of motion verbs. I describe the classifier model and the selection of model features, and I perform an error analysis on the results. Finally, I test the effect of figurative/literal annotations of English manner of motion verbs on an existing machine translation system.

2. Data

2.1 Source

The data for the classifier were 2200 tokenized English sentences from Europarl, all containing a manner of motion verb as classified Levin (1993)—see Table 1 for the full list. These sentences were automatically selected from a set of 1M English Europarl sentences if they contained a word in Table 1 (or a derivation thereof) that was tagged as a verb by the Stanford Part-of-Speech Tagger.

All Manner of Motion Verbs
amble, bolt, bounce, bound, bowl, canter, carom, cavort, charge, clamber, climb, clump, coast, crawl, creep, dart, dash, dodder, drift, file, flit, float, fly, frolic, gallop, gambol, glide, goosestep, hasten, hike, hobble, hop, hurry, hurtle, inch, jog, journey, jump, leap, limp, lollop, lope, lumber, lurch, march, meander, mince, mosy, mosey, nip, pad, parade, perambulate, plod, prance, promenade, prowl, race, ramble, roam, roll, romp, rove, run, ran, rush, sashay, saunter, scamper, scoot, scram, scud, scurry, scutter, scuttle, shamble, shuffle, sidle, skedaddle, skip, skitter, skulk, sleepwalk, slide, slink, slunk, slither, slog, slouch, sneak, snuck, somersault, speed, stagger, stomp, stray, streak, stride, stroll, strut, stumble, stump, swagger, sweep, swim, tack, tear, tore, tiptoe, toddle, totter, traipse, tramp, travel, trek, troop, trot, trudge, trundle, vault, waddle, wade, wad, walk, wander, whiz, zigzag, zoom
Verbs Excluded due to Competing Semantic Senses
bound, charge, file, hasten, hike, nip, pad, ran, shuffle, skip, speed

Table 1: Manner of Motion Verbs from Levin (1993)
Levin (1993)'s verbs of motion around an axis were not included. Some verbs were excluded if they had a very common sense besides manner of motion (e.g., one can be *bound to a proposal*, one can *file paperwork*, or one can *run a company*), in order to simplify the classification task.

2.2 Figurative/Literal Annotation for Supervised Learning

Manner verbs in the Europarl sentences were classified as literal or figurative via a task posted to Amazon's Mechanical Turk. Workers on Mechanical Turk saw a single sentence at a time and were asked to categorize the verb (presented in all caps) as figurative or literal. They were instructed that literal words "describe a real physical event" and "involve physical movement by a real person, animal, or thing," while figurative words "describe a conceptual event" and "will not involve real physical movement of a real person, animal, or thing." Each sentence was presented to three workers, and classified according to majority vote. The Mechanical Turk annotations had 95.4% agreement with 124 sentences hand-annotated by the author.

2.3 Data Sets

The 2200 sentences were randomized and partitioned into three sets: a set of 1800 training sentences, a set of 200 development (dev) sentences, and a set of 200 held-out test sentences for the final evaluation. The breakdown of these sets is listed in Table 2 below (the statistics for the test set were not calculated until after the classifier performance on the test set was analyzed).

	Literal	Figurative	Total
Training set	731	1069	1800
Dev set	84	116	200
Test set	66	134	200
Total	*881*	*1319*	*2200*

Table 2: Counts of sentence classes in the three data sets

As the above table shows, the training data is imbalanced between the classes, with 59% of its sentences annotated as figurative. However, given the small size of the total annotated data set, I chose to train on imbalanced input data to avoid further diminishing my training set size.

3. Classifier Model

To distinguish between figurative and literal uses of manner verbs, I implemented a linear classifier with a logistic loss function (much of the code for which was written in CS221), which distinguished between two classes: -1 (figurative) and 1 (literal). Each feature f_i was assigned a unique weight w_i, which were optimized via stochastic gradient descent to minimize the total logistic loss over each training sentence x in class c:

$$\text{LogisticLoss}(x) = \text{Log}[1 + \text{Exp}[-c*\textstyle\sum_i w_i*f_i(x)]]$$

The classifier performed only three rounds of stochastic gradient descent to prevent overfitting on the small training set. It then predicted the class of each example according to the following:

$$\text{Class}(x) = \text{Sign}(\textstyle\sum_i w_i*f_i(x))$$

4. Description of the Classifiers

I used the linear classifier model above to build two different figurative/literal classifiers: a baseline bag-of-words system and the real classifier with more carefully designed features.

4.1 Bag-of-words Classifier

The bag-of-words had a feature for each word string that appeared in the training data, whose value for a particular sentence was equal to the number of times that the word appeared in the sentence. Thus, for a sentence x and word w, $f_w(x) = \text{Count}(w \text{ in } x)$

4.2 Final Classifier

The final classifier included four sets of features, listed below:

4.2.1 Verb Lemma Features
These were simple binary features VERB:*lemma* that captured the lemma of the manner verb in the sentence.

4.2.2 Dependencies Type Features
I included set of features DEP:*type* that capture the syntactic dependency relation type of each direct dependent of the manner verb, according to a Stanford Dependencies collapsed dependencies parse generated by the Stanford Parser. These were binary features: for example, even if a verb had two adverbial modifier dependents (as in the phrase *walk quickly and efficiently*), the value of DEP:*advmod* for that sentence would be 1.

4.2.3 WordNet Hypernyms – Lower
I added a set of features HYP-LOWER:*synset* intended to capture lower level semantic categories to which noun dependents belonged. For each noun dependent of the manner verb (determined by the part of speech assigned by the parser), a set of lower-level hypernyms was generated via WordNet by tracing upwards two levels in the WordNet hierarchy of hypernyms from the WordNet Synsets of the noun[1]. Since I did not perform any word sense disambiguation on the noun dependents, I generated the set of all hypernyms two levels up from all the Synsets associated with the noun and created a binary feature for each hypernym in the set.

4.2.4 WordNet Hypernyms – Higher
Finally, I included a set of features HYP-HIGHER:*synset* intended to the capture higher level categories of noun dependents on the manner verb. All nouns in WordNet have the 'entity' Synset as their highest hypernym. I generated the set of inherited hypernyms of the noun that were two levels below the 'entity' Synset and created a binary feature for each.

5. Bag of Words Classifier

The bag of words classifier was intended to provide a baseline performance metric against which to evaluate the final classifier.

5.1 Performance

Table 3 below shows the performance of the final classifier on the dev and test sets. It includes total error (% of examples classified incorrectly), as well as precision, recall, and F1 scores for each class.

	Dev set	Test set
Error	**0.245**	**0.225**
Lit. Prec.	0.697	0.621
Lit. Rec.	0.738	0.818
Lit. F1	**0.717**	**0.706**
Fig. Prec.	0.802	0.893
Fig. Rec.	0.767	0.754
Fig F1	**0.7841**	**0.818**

Table 3: Performance of the classifier on the development and test sets

The simple bag of words classifier performed relatively well, suggesting that a classifier with better tuned features than bag of words could be even more successful.

5.2 Error Analysis

The bag of words classifier committed errors because many words were given weights that reflected arbitrary features of the training set rather than true information about what distinguished between literal and figurative expressions. For example, a period (.) was assigned a negative weight of -.89 (likely because it occurred in nearly all sentences, and there were more figurative examples than literal ones), while a dash (-) was assigned a positive weight of 0.69. In all likelihood, punctuation does little to actually determine whether an expression is

[1] A WordNet Synset (synonym set) is a set of immediate synonyms of a particular sense of a word. A single word will have multiple Synsets, reflecting its different semantic senses. For example, the noun *club* has the following Synsets, among others: {Synset('clubhouse.n.01'), Synset('golf_club.n.02'), Synset('cabaret.n.01'}. A Synset in WordNet is also associated with a set of hypernym (or superclass) Synsets; for example, the direct hypernym of Synset('clubhouse.n.01') is Synset('building.n.01'). These sets of hypernym relations form a hierarchy of Synsets from each noun to the ultimate noun hypernym, *entity*.

literal or figurative. The weights given to punctuation as well as other uninformative words resulted in incorrect classifications, such as in the (somewhat gruesome) example below from the test set

- Example: *The man managed to walk a couple of kilometres, killed his wife and threw her body out of the window.*
    - True class: literal
    - Assigned class: figurative
    - Explanation: Several uninformative words contributed negative feature weights to result in an incorrect classification. In addition to a period, these words include: *a* (weight = -0.33), *his* (weight = -0.82), *the* (weight = -0.15) and *and* (weight = -0.09).

Thus, a more refined feature set was required to reduce the number of spurious features in the classifier.

6. Final Classifier

6.1 Performance

Table 4 below shows the performance of the final classifier on the dev and test sets. It includes total error (% of examples classified incorrectly), as well as precision, recall, and F1 scores for each class.

	Dev set	Test set
Error	**0.145**	**0.140**
Lit. Prec.	0.831	0.797
Lit. Rec.	0.821	0.773
Lit. F1	**0.826**	**0.785**
Fig. Prec.	0.872	0.890
Fig. Rec.	0.879	0.903
Fig F1	**0.875**	**0.896**

Table 4: Performance of the classifier on the development and test sets

The final classifier performed better on all counts than the bag-of-words classifier. It showed a bias toward the majority figurative class on both the dev set and the test set. This is likely an effect of the imbalanced training data, and of the small size of the training set in general. The classifier performed better on the literal class in the dev set than in the test set. The higher figurative recall of the classifier on the test set represents a direct tradeoff with literal recall, which is much lower on the test set. The classifier's worse performance for literal on the test set may have been related to the smaller fraction of literal sentences in the test set (33%, compared to 42% in the dev set and 41% in the training set). The difference also could have resulted from a different distribution of verbs in the test set, though examining distribution of the top 10—see Table 5 below for the top 5 in each set. Though Table 5 does not reveal a clear pattern, the lower percentage of figurative uses of *fly* in the test set than the dev and training sets may have contributed.

Training set			Dev set			Test set		
Top 5	% of set	% fig.	Top 5	% of set	% fig.	Top 5	% of set	% fig.
travel	23	7	travel	22	11	travel	19	13
fly	8	43	rush	9	78	roll	7	93
walk	7	51	walk	8	56	rush	7	79
roll	7	90	fly	7	23	fly	6	17
rush	6	70	float	6	75	walk	6	64

Table 5: Distribution of the top 5 verbs in the training, dev, and test sets. Shown are the percent of sentences in the set including the verb and the percent of those sentences that were figurative.

Since the goal of the classifier was to split apart figurative and literal instances of each individual verb, I also calculated the classifier's performance on the verbs *walk* and *fly*, chosen because they were common in the training set and had a relatively even distribution of literal and figurative uses. Table 6 below shows the classifier's performance on these verbs.

	Walk		Fly	
	Dev set	Test set	Dev set	Test set
Counts	*16 (7 fig.)*	*11 (7 fig.)*	*13 (3 fig)*	*12 (2 fig.)*
Error	**0.125**	**0.091**	**0.231**	**0.167**
Lit. Prec.	0.778	0.800	0.818	0.900
Lit. Rec.	1	1	0.900	0.900
Lit. F1	**0.875**	**0.889**	**0.857**	**0.900**
Fig. Prec.	1	1	0.500	0.500
Fig. Rec.	0.778	0.857	0.333	0.500
Fig. F1	**0.875**	**0.923**	**0.400**	**0.500**

Table 6: Performance of the classifier on the verbs *walk* and *fly*.

The classifier performed quite well for *walk* on both the dev set and the test set, but struggled to correctly classify figurative instances of *fly*. The poor performance on *fly* is likely related to the small number of figurative *fly* examples in the dev and test sets, which magnified the effect of a single misclassification. Additionally, many of the examples with *fly* may have been difficult for Mechanical Turk workers to classify as figurative or literal, as they involved instances of countries or companies *flying flags* of various nations, which could refer to literal flags or figurative flags. In any case, the results of the classifier on *walk* are encouraging, as they suggest that with more training examples of the less common verbs and with a more balanced data set, the accuracy of the classifier would increase.

6.2 Design

The following section describes my rational for including each of the four feature sets in the classifier, as well as a discussion of feature sets that were tried but abandoned.

6.2.1 Verb lemmas

I added the verb lemma feature to capture the tendency of a particular verb to be figurative or literal. For example, *travel* was literal 93% of the time in the training data, while *roll* was figurative 90% of the time, and the classifier should take these tendencies into account. With this feature alone, classification error on the dev set was 0.205, with literal F1 of 0.725 and figurative F1 of 0.837, equivalent to assigning each verb to its majority class. The other classifier features, described below, were designed to then make distinctions between verbs with the same lemma.

6.2.2 Dependency types

I next added binary features capturing the syntactic dependency relation type of each direct dependent of the manner verb. The logic behind these features is that figurative and literal instances of the same verb likely have different argument structures. In particular, kinds of prepositional phrases attached to the verb may differ depending on whether it is being used literally or figuratively. For example, *walk* ***into*** *the negotiations* is usually a figurative phrase, describing the process of starting negotiations, whereas *walk* ***to*** *the negotiations* is more likely to describe the literal way in which someone travelled to the site of the negotiations. I thus chose to capture details about the syntactic structure of the manner verb phrase with a Stanford Dependencies collapsed dependencies parse, which directly encodes the type of syntactic relations between a verb and its dependents tags objects of prepositions as directly dependent on the verb, with the preposition encoded in the dependency type.

Three dependency types were excluded from this feature set: *nsubj* (noun subject), *aux* (auxiliary verb), and *punct* (punctuation). These dependencies were very common in the data set across the two classes, but because there were more figurative training examples available than literal ones, they received a negative weight, leading to classification errors. Thus, excluding them increased the performance of the classifier.

The dependency type features alone produced a classification error of 0.270 on the dev set, with literal F1 of 0.640 and figurative F1 of 0.748. The adding the dependency type features to the verb lemma features decreased the error to 0.195 on the dev set, with literal F1 of 0.745 and figurative F1 of 0.842.

6.2.3 Hypernyms

The hypernym features were designed to capture semantic information about the dependents of the manner verb, as much of the information that distinguishes literal from figurative expressions comes from their semantic content. I added two types of hypernym features: a set of hypernyms near the bottom of the hypernym hierarchy, and a set of hypernyms near the top. The higher hypernyms include Synsets with labels such as *group*, *object*, *attribute*, and *causal agent*, having the potential to capture generalizations such as, for example, that the kinds of things that literally move (e.g. causal agents), and the kinds that only move in a figurative sense (e.g., attributes). The lower hypernyms, on the other hands, were intended to capture generalizations of more specific types of semantic arguments to the verb. For example, while physical *objects* may frequently be the arguments of verbs in literal expressions, certain physical objects are frequently referred to in a figurative sense, such as *ceiling* in *glass ceiling*. I chose to use hypernyms of the noun arguments rather than the argument themselves, however, in order to better generalize over the small training set.

Adding just the higher hypernyms to the verb lemma and dependency type features increased classifier performance to 0.170 error on the dev set, with literal F1 of 0.795 and figurative F1 of 0.855. Adding just the lower hypernyms to the verb lemma and dependency type features increased classifier performance to 0.160 error on the dev set, with literal F1 of 0.803 and figurative F1 of 0.866.

6.2.4 Unused Features

I was unable to successfully model the interactions between the verb, its dependency types, and the semantic classes of its arguments (represented in my features as hypernyms). Intuitively, there should be interactions between all these sets of features, as in the examples listed below. For example, if we say that *the delegation stumbled*, we are probably using a figurative expression meaning that the delegation made a mistake. However, if we say that *the delegation flew*, we probably mean that the members literally took a plane. In contrast, both *the man stumbled* and *the man flew* seem likely to be literal. Thus, the same semantic argument type interacts with the verb. Likewise, the dependency type and the semantic content of the dependent should interact: if we say *the negotiations trudged past*, with *negotiations* as the subject, we are likely using a figurative expression to mean that the negotiations moved slowly. However, if we say *the man trudged past the negotiations*, we probably mean that the man literally moved past the site of negotiation.

I tried adding interaction features for dependency type + higher hypernym, dependency type + lower hypernym, verb lemma + higher hypernym, and verb lemma + lower hypernym. However, the addition of each of these interaction features worsened the performance of the classifier. This was most likely due to the small size of the training set compared to the feature space of the interaction features: with 127 manner of motion verbs, hundreds of collapsed dependency types (the exact count is not known due to the process of collapsing prepositions), and hundreds or thousands of lower hypernym classes, combining the features did not produce generalizable feature weights.

6.3 Error Analysis

Several common error types emerged in the classifier output for the test and dev sets, falling into two major categories: mistakes caused by the model and features used, and mistakes caused by noise in the data set. Common types from both categories of features are listed in the sections below. In all examples, the manner verb has been highlighted with all caps.

6.3.1 Errors Caused by the Model

The following errors were caused by the model and feature design.

1. Few syntactic dependents: When the motion verb had few or no syntactic dependents, the verb lemma itself became the primary feature that determined the output of the classifier. Thus, the sentence was classified according to the most commonly seen class for that verb in the training data, which was incorrect for an important minority of examples.
    - Dev set example: *If the people speak, if we allow uncensored access to the internet, I can tell you that it will be a great moment for the Olympic games, for sport and for democracy, because sport and democracy must go hand in hand; otherwise, there is no point JUMPING, running or swimming.*
        - True class: literal
        - Assigned class: figurative
        - Explanation: The dependency parse assigns only one dependent to *jumping*: *swimming* is considered a *conj_or* dependent. Thus, the classifier simply relies on the negative feature weight of jump (weight = -0.99) to classify the sentence.
    - Test set example: *Despite the commitment they have shown to the company and despite their high level of professionalism – which has made the cisterna plant one of the most productive in Europe – these workers are likely to be removed from the production cycle, while Goodyear, after years of benefiting from state and European subsidies, is totally free to WALK out.*
        - True class: figurative
        - Assigned class: literal
        - Explanation: The dependency parse assigns only one dependent to *walk*: *out* is a verb particle dependent. Thus, the classifier relies on the positive feature weight of *walk* (0.66) to produce an incorrect literal classification.

2. Hypernym overload: Because I included all higher and lower hypernyms of all Synsets of a single noun dependent as features, many sentences had 10-15 hypernym features. Since I performed no word sense disambiguation, the set of hypernyms often contained a number of abstract concepts as well as a number of concrete entities. A large set of hypernyms could thus lead to the wrong classification if too many hypernyms from the wrong sense were included in the set.
    - Dev set example: *Thick foam FLOATS on the surface of the river.*
        - True class: literal
        - Assigned class: figurative
        - Explanation: The dependency parse of the sentence correctly tagged *foam* and *surface* as dependents of *floats*. There were 13 higher and lower hypernyms associated with these nouns, of which three were abstract concepts with large negative weights (*psychological_feature*, *feature*, and *instrumentality*), resulting in a figurative classification.
    - Test set example: *They believe that the mind to negotiate will melt away, especially as airlines from the US already have de facto stand alone cabotage to FLY from one member state to another, which we can not do on US territory.*
        - True class: literal
        - Assigned class: figurative
        - Explanation: The dependency parse correctly assigned *state* and incorrectly assigns *mind* as noun dependents of *fly*. Together, these two nouns have 18 higher and lower hypernyms, 11 of which have negative weights, resulting in a figurative classification.

3. Verb majority class bias: Certain verbs had very large positive or negative weights associated with their lemma, as they fell into a particular class in nearly all instances in the training data. These large weights overrode the contributions from other features. This was perhaps the most common error committed by the classifier, and happened especially frequently with the verb *travel*.
    - Dev set example: *Both Turkey and the EU have a very long road to TRAVEL, politically and economically, before Turkish membership can be an imminent possibility.*
        - True class: figurative
        - Assigned class: literal
        - Explanation: Literal uses of *travel* occurred so frequently in the training set that the feature VERB:*travel* received a weight of 3.01 the largest weight of all the features (by absolute value). The sentence had 8 other features, 6 of which had negative weights; however, this was not enough to overcome the literal bias of *travel*.
    - Test set example: *Nobody argues against Russia's right to TRAVEL down a specific path, if the majority of citizens want this.*
        - True class: figurative
        - Assigned class: literal
        - Explanation: As above, *travel* has such a large positive weight, outweighing all of the 7 other features, 6 of which have negative weights.

4. Figurative arguments: In figurative language, sometimes a reference to a concrete object is made in a figurative sense. For example, in the phrase *climb the ladder of success*, a *ladder* is a concrete object but is being referred to figuratively. However, the features of the classifier did not catch this kind of figurative usage, as concrete objects were usually referred to literally in the training data and thus the feature weights for concrete hypernyms were generally positive.
    - Test set example: *And the man who ordered the killing still STRUTS the streets of Belfast in the company of the IRA's chief of intelligence, Bobby Storey, and leading provisional IRA man, Eddy Copeland.*
        - True class: figurative
        - Assigned class: literal
        - Explanation: the parser identifies *man* and *streets* as dependents of *strut*. Together, these nouns have 18 higher and lower hypernyms, most of which are concrete objects, and 14 of which have positive feature weights, resulting in a literal classification.

6.3.2 Errors caused by noisy data

The following errors were caused by noise and other issues with the input data set:

1. Incorrect parse: The parser sometimes produced the wrong parse of a sentence, leading to an incorrect classification. This occurred four times in the test set and once in the dev set (I did not quantify the number of parse errors in the training set, as this would have been quite time consuming).
    - Training set example: *It is as if their own journeys are invaluable but others TRAVEL just for fun.*
        - True class: literal
        - Assigned class: figurative
        - Explanation: The parser incorrectly identified *journeys* as a verb, which was then selected as the manner verb rather than *travel*. Though *journey* has a positive feature weight, it is not positive enough to overcome the negative weights of other features of the sentence. However, the large positive feature weight of *travel* would have resulted in a correct classification.

- Test set example: *As far as the present situation is concerned, on 28, 29 and 30 March a commission delegation will TRAVEL to Malta to discuss the case in greater detail.*
    - True class: literal
    - Assigned class: figurative
    - Explanation: The parser incorrectly identifies *March* as a verb, which is selected as the manner verb over *travel*. As above, had travel been identified as the verb, a correct classification would have resulted.

2. Incorrect classification on Mechanical Turk: Quality control is hard to maintain on Mechanical Turk, and occasionally the workers produced the wrong annotation. Although comparing the Mechanical Turk annotations to my hand annotations produced only 4.6% error from Mechanical Turk, in practice error may have been larger on the entire data set. The test set in particular appeared to have quite a few of these errors (though I did not tabulate the exact number, as these judgments are somewhat subjective).
    - Dev set example: *I congratulate her on such an initiative but, at the same time, regret the fact that the commission is being called on yet again to close the door after the horse has BOLTED instead of preventing such unfortunate incidents.*
        - Turker annotation: literal
        - True class: figurative
        - Explanation: The workers were instructed to classify words as literal only if they described physical movement of real physical objects. Here, there is no real *horse*—instead, *bolted* is being used as part of a figurative expression, but this was not caught by the Turkers.
    - Test set example: *I believe that he deserves a lot of credit: he has carried out his mission under extremely difficult conditions; with his considerable experience he has succeeded in seeing things in context when it has been necessary not to be absolutist, in being severe when it was necessary to be severe, and, in any case, in remaining rigorous and never STRAYING from his course.*
        - Turker annotation: literal
        - True class: figurative
        - Explanation: Though the sentence does not describe a physical movement event in which someone actually *strays* from a path or course, workers nevertheless classified the sentence as figurative.

3. Ambiguous examples: In designing the Mechanical Turk experiment and building the classifier, I made the simplifying assumption that all data can be categorized as figurative or literal. However, in actuality, the distinction between figurative and literal expressions is not always clear-cut, making some sentences difficult to annotate and the output of the classifier on these sentences difficult to analyze. The test set in particular appeared to have a quite a few such examples, though whether a sentence is truly ambiguous between figurative or literal is a highly subjective judgment.
    - Training set example: *Mr. President, thanks to science and technology, mankind has begun to conquer space and has WALKED on the moon.*
        - Turker annotation: figurative
        - True class: ??
        - Explanation: It is not accurate to say that all of mankind has literally walked on the moon. However, this sentence does in some sense reference a literal walking event, making it difficult to classify.

- Test set example: *Let us think of those who have started the painstaking process of CLIMBING the stairs.*
    - Turker annotation: literal
    - True class: ??
    - Explanation: Due to lack of context, it is impossible to tell whether the speaker refers to a literal set of stairs (perhaps he or she is discussing people in a rehabilitation process) or a figurative set of stairs (perhaps signifying the rebuilding of a town or nation).

6.4 Time and space complexity

Statistics about the time and space complexity of the classifier are listed in Table 7 below. Here, S represents the number training sentences, T the number of sentences to be classified, D the maximum number of dependents of a single manner verb, D_{tot} the total number of dependency relation types, H the maximum number of hypernym types (higher or lower) of a single noun dependent, H_{low} the total number of lower hypernym types, H_{high} the total number of higher hypernym types, and V the total number of manner verb lemmas

Training time	O(SDH)
Classification time	O(T(D+H))
Space Complexity	$O(V+D_{tot}+ H_{high}+ H_{low})$

Table 7: Space and Time complexity of Linear Classifier. Time complexity is split into two categories: training time (generation of the feature weights) and classification time (classification of the dev or test set).

The actual (user + system) runtime of the classifier, measured with the Unix *time* command on the test set, was only 5.174 seconds. Of this, 1.992 seconds were consumed by loading the featurized sentences with the Python pickle module, and 2.98 seconds were consumed by generating the feature weights (leaving only 0.202 seconds of classification time). In part due to the small size of the data set, the classifier ran quickly. The main slowdown in the classification process was parsing sentences with the Stanford Parser, which took between 1 and 2 seconds per sentence (requiring almost an hour to parse the entire data set). A simple solution to this slowdown would be to parse with MaltParser, which parses to a dependency representation in linear time.

6.5 Future Directions

While the classifier performed relatively well given the small size of the training data, the most straightforward improvement would be to create a larger, more balanced annotated data set with many instances of each manner of motion verb. Given the promising results of the classifier on *walk*, a larger, more balanced data set would likely improve performance for each individual manner of motion verb. Additionally, a larger data set would allow the addition of interaction features, which could further improve performance.

Besides expanding the data set, the classifier could be improved by refining the feature set to target areas where the classifier made mistakes. In particular, the classifier has little information to deal with sentences where the manner verb has no dependents. In these cases, adding features with information about the head verb of the sentence, the noun subject of the sentence, or all the nouns in the sentence might improve performance (although, in turn, this would make the feature space even larger and necessitate more training data). It might also help to produce a dependency parse with propagation of conjunct dependencies, which would reveal dependents of the manner verb that are currently not shown in cases of conjunction. Lastly, the feature set could be refined by only including as features hypernyms that occur above a certain frequency in the training data, to prevent hypernym overload.

7. Machine Translation System Evaluation

As my inspiration for building the figurative/literal classifier was to improve the performance of machine translation on English manner verbs, I tested the effect of annotating these verbs as figurative or literal before inputting them to the Moses open source statistic machine translation system.

7.1 Data

The data sources used as inputs to Moses are listed below.

- **Alignment corpora**: 50000 parallel French and English sentences from the Europarl corpus, aligned with the GIZA++ word aligner. Of these, 8129 sentences contained string matches to a manner of motion verb (though not all contained a manner of motion verb). This data set contained some of the training data for the classifier.
- **Tuning data**: 1000 parallel French and English sentences from the Europarl corpus (no overlap with the alignment corpora). All tuning sentences contained a string match to a manner of motion verb, and 381 sentences contained a manner of motion verb, according to their dependency parse. This data set contained some of the training data for the classifier.
- **Evaluation data**: 1000 parallel French and English sentences from the Europarl corpus (no overlap with the above data sets). All evaluation sentences contained a string match to a manner of motion verb, and 341 sentences contained an actual manner of motion verb, according to their dependency parse. This data set contained some of the training data for the classifier.
- **Language model**: 3M English sentences from the Europarl corpus (including all the sentences from the alignment corpora). I made the error of not checking the language model data against the tuning and testing data, so there may have been overlap with these data sets. However, this was true for both the annotated data and the unannotated, so the results should still reveal the effect of adding annotation. Nevertheless, optimally the experiment should be rerun with this error fixed.

7.2 Experiments

I performed two runs of the MT system. On the first run, I performed no annotation of the manner of motion verbs, and simply input the data sets directly to Moses. On the second run, for all data sets, I ran all sentences containing a string match to a manner of motion verb through the classifier. If the string match was not classified as a verb in the dependency parse, I performed no annotation. Otherwise, I replaced the manner of motion verb *mverb* as *lit:mverb* or *fig:mverb* in the sentence, depending on the output of the classifier. I ran Moses with a maximum phrase length of 4, a distortion limit of 8, and MERT tuning, as these parameters produced the best performance in my PA1 assignment.

7.3 Results and Discussion

A total of 663 of the unannotated output sentences contained a string match to a manner of motion verb, and 652 of the annotated output sentences contained a string match to a manner of motion verb (roughly equivalent). Table 8 below compares the BLEU scores of the Moses system run with annotated and unannotated data.

	Annotated	Unannotated
Unigram score	64.2	61.3
Bigram score	36.0	29.2
Trigram score	22.9	16.1
4-gram score	15.1	9.4
Total score	***28.99***	***22.22***

Table 8: Comparison of BLEU scores on annotated and unannotated inputs to Moses. Annotated inputs had manner of motion verbs tagged with the prefix *fig:* or *lit:*.

The annotated input performed better on all counts than the unannotated input. However, certain complications prevent concluding that distinguishing between figurative and literal manner of motion verbs was the direct cause of this improvement.

One complication was that the literal/figurative amounted to giving Moses information on whether or not a word was a verb, as in the examples sentences below (emphasis mine):

- Target sentence (annotated): *the erika , a vessel **lit:flying a maltese flag** , a floating rust bucket , classed amongst the most dangerous type of oil tankers , has contaminated more than 400 km of our coastline , a case of pollution even worse than that caused by the amoco cadiz .*
- Annotated output: *the erika , ship **lit:flying a maltese flag** , the tanker wreck classified among the most dangerous , ends on our coasts more than 400 km , which represents a much more serious than pollution caused by the amoco cadiz.*
- Unannotated output: *the erika , ship **flag-of-convenience maltese** wreck classified among the oil tankers the most dangerous , has ends our coasts on more than 400 km , this which represents a pollution much more serious as that caused by amoco cadiz .*

Adding the *lit* tag helped Moses recognize that the sentence required the verb *fly*. While adding part of speech information clearly helps the system, this improvement is not related to the literal/figurative distinction. An improvement would be to add the tag *verb* to manner of motion verbs in the unannotated corpus.

Another complication was that the annotated data appeared to be generated with an overall better language model, as exemplified in the sentences below, which did not contain a manner of motion verb (emphasis added to highlight more natural annotated output).

- Target sentence: *you addressed plenary in strasbourg nearly six years ago , in march 1994 , and **many of us** , who were already european members of parliament at that time , still have a vivid recollection of your speech .*
- Annotated output: *you have already spoken in plenary in strasbourg , there is almost six years , in march 1994 , and **many of us** to retain a very strong memory , i.e. those of our fellow members who were already at that time .*
- Unannotated output: *you have already discussed in plenary , in strasbourg , it nearly six years , in march 1994 , and **many between we** retain a remember very strong , i.e. of our colleagues who were already parliamentary to this era .*

The annotated and unannotated language models were both compiled with the same set of Europarl sentences. However, it would be worth rerunning the experiment with regenerated language models, in case an error (human or otherwise) occurred in their original generation. Alternatively, the better information about part of speech, figurative usage, and literal usage could have resulted in the better language model.

Despite these complications, the literal/figurative tagging does appear to make a difference for certain sentences. The Moses translations successfully distinguish between literal and figurative verbs in cases such as these:

- Annotated output sentence 1: *the trial against journalists **fig:fly** in the face of all legal principles .*
- Annotated output sentence 2: *we intend to turn this situation , especially given that security is sometimes in danger , for example in cases where the pilots to **lit:fly** too long .*

And this distinction can result in translations that are more similar to the target, as in the sentences below (same example as output sentence 2 above).

- Target sentence (annotated): *the legal proceedings against the journalists **fig:fly in the face** of every legal principle .*
- French source sentence: *les procès intentés contre les journalistes **vont à l' encontre** de tous les principes juridiques .*
- Annotated output: *the trial against journalists **fig:fly in the face** of all legal principles .*
- Unannotated output: *trials against journalists **run counter** of all the legal principles .*

While the French *vont à l' encontre* literally translates to something more similar to *run counter to*, the figurative/literal tagging helps the Moses system associate the verb *fly* with the French, producing a better target sentence output.

Despite the complicating factors in the Moses experiment, results such as the sentences described above suggest that literal/figurative tagging does contribute to better BLEU scores. To verify this conclusion, though, a second Moses experiment should be conducted. In this experiment, the language model should be regenerated, and the data used to build it should be guaranteed to have no overlapping sentences with the tuning and test data. The data that is not annotated for the figurative/literal should instead have all manner of motion verbs tagged as *verb*. Such an experiment would produce clearer results about the effect of figurative/literal tagging.

8. Conclusion

With a few automatically derived features modeling the semantic and syntactic environment of English manner of motion verbs, I was able to construct a relatively high accuracy classifier for distinguishing between literal and figurative verb uses, with only 14% error on my test set, and F1 scores of 0.773 and 0.903 on the literal and figurative data, respectively. However, the classifier has plenty of room for improvement, especially in its performance on literal verbs, and its performance could be increased with a larger data set and more carefully tuned features. My experiments with Moses suggest that including information about figurative/literal distinctions can indeed improve the performance of a machine translation system, but further experimentation is required to refine and verify my results.

9. References

Birke, J. and A. Sarkar (2006). A clustering approach for the nearly unsupervised recognition of nonliteral language. In In Proceedings of EACL-06, pp.329–336

Fellbaum, C. (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

de Marneffe, M., B. MacCartney and C. Manning (2006). Generating Typed Dependency Parses from Phrase Structure Parses. LREC 2006.

Gedigian, M., J. Bryant, S. Narayanan, B. Ciric (2006) Catching metaphors. Proceedings of the Third Workshop on Scalable Natural Language Understanding, New York City.

Klein, D. and Manning, C. (2003) Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.

Koehn, P. (205). Europarl: A Parallel Corpus for Statistical Machine Translation. MT Summit 2005.

Koehn, P., et al (2007). Moses: Open Source Toolkit for Statistical Machine Translation. Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic.

Levin B. (1993) English Verb Classes and Alternations: A Preliminary Investigation. Chicago: University of Chicago Press.

Mason, Z. (2004). Cormet: a computational, corpus-based conventional metaphor extraction system. Computational Linguistics, 30(1):23–44.

Miller, G. (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

Och, F. and H. Ney (2003). A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, 29(1):19-51.

Peters, W. and I. Peters (2000). Lexicalised systematic polysemy in wordnet. In Proceedings of LREC 2000, Athens.

Shutova, E. 2011. Computational Approaches to Figurative Language. PhD thesis, Computer Laboratory, University of Cambridge, UK.

Toutanova, K., D. Klein, C. Manning, and Y. Singer (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. Proceedings of HLT-NAACL 2003, pp. 252-259