Lack of Spatial Indicators in Hamlet

Christine Talbot and **G. Michael Youngblood**
Game Intelligence Group
University of North Carolina at Charlotte
{ctalbot1, youngbld}@uncc.edu

Abstract

While researching spatial movements in play-scripts, we uncovered some movements that actors performed that could not be explained by the annotations or basic theatre rules. Here, we look to learn implied motion based on what the characters in the play are saying. Humans are able to do this with only being given the play-script, so how do we get a computer to do it?

Several features, including n-grams, parts of speech (POS) bag of words (BoW), length of speech, and other contextual details were utilized with several machine learning methods to help predict movement within the play. Results reveal that is a difficult problem and basic natural language processing (NLP) and machine learning (ML) techniques do not perform much better than a random classifier.

Introduction

When actors perform on stage, they are provided with specific directions on where and how to perform their lines. The director provides these directions via a play-script's annotations. Beyond these annotations, the actors are provided some freedom in performing their lines, although certain guidelines for theatre acting are always in play. Intuition and characterization help the actor to identify other movements that are in-character and appropriate in the different parts of the play for their character.

We look to realistically capture the spatial movements of actors on stage, so we started by translating the spatial movements found within the annotations from the director, as can be seen in our prior work (Talbot and Youngblood 2012). Basic parts of speech (POS), sentence structure parsing, and entity recognition provided us with key movements detailed from the annotations in the play-script with about 75% accuracy for character positions.

Next, we targeted the basic rules and guidelines that actors and directors use to control movement on the stage (Talbot and Youngblood 2013). These included conversational space, group space, theatre rules, and general common-sense rules. This provided 89% accuracy for position and 53% accuracy for gaze. After capturing these movements, there were still some movements in the play which the actors performed, but were not captured by the annotations and rules we encoded. One good example is in Act V in Hamlet where the gravedigger walks towards the audience, then turns around and walks back towards the grave. These are the kinds of movements that the actor decides upon based on their intuition.

Therefore, we thought about what might help a system to learn these sort of movements by the actors. We came to the hypothesis that perhaps what the actor is saying could imply certain types of movement. Now, these are not the same kinds of movements as one actor telling the other actor to do something, but more of an implied movement, such as moving towards the audience for a monologue, gesturing to help emphasize what they are saying, or even a movement to keep the audience's attention during a rather long scene that has little to no movement involved with it. We are not focused on what is explicitly stated in the language, but more on the hidden movement that is likely to be performed by the actor on stage.

The context of the speech and the characters were identified as two key components to interpreting the implied movement, in addition to what the character is saying. We pursued several existing natural language processing and machine learning approaches to learn these implied movements within one particular play, Hamlet, as produced by Sir Gielgud in 1964 on Broadway (Colleran et al. 1964). We utilized the script as written by Shakespeare (Shakespeare 1905), as well as the Electronovision video (Colleran et al. 1964) of Richard Burton in Sir Gielgud's production of Hamlet as our baseline. Each line's related movement was captured for the play and categorized into a standardized set of motions, such as walking, jumping, fighting, and so forth. We fed this information into machine learning algorithmsto help learn about the implied movements within the play.

Naturally, while pursuing an appropriate approach for our work, we started with the natural language processing that is used for giving directions to robots. This incorporates both natural language and spatial reasoning. However, the key difference with what we were looking to do is that we were not trying to give explicit directions for someone to do something. We want to understand the hidden movement. So looking at work like Wei et al.'s (2009), Brooks' (2012), and Kollar et al.'s (2010) only provided techniques that assumed a set of predefined keywords, phrases, or corpus to be extracted and utilized for further processing. These focused on the meaning of different prepositions in order to interpret a spatial location.

Next, we looked into text categorization and summarization. The main focus of most text categorization is around known keywords or phrases to identify if the text contains that concept. The more similar the strings or synonyms are, the more similar they are considered to the entity being matched. The summarization techniques, like those used in Chuang and Yang's (2000)paper, focus on segmentation of the text and the extraction of important sentence segments via machine learning on a feature vector. This is closer to what we want to do, but still is based on keywords and phrases, with little to no implied meaning involved.

A main exception to the patterns of text classification was with the data-less categorization done by Chang et al. (2008). They focused on the semantic meaning of the category to determine how to classify text without labeling and training the system. Also, classifying text into multiple categories is still not completely solved, as discussed in Platt, Cristianini, and Shawe-Taylor's (2000) paper. This is key as we look at our data where one line can imply more than one motion. Some researchers, such as Schapire and Singer (2000), have pursued multiple class classifications by using Boosting and text classification where the problem is not turned into multiple binary classification problems, as is typical for this problem.

Other work with ConceptNet (Liu and Singh 2004) also is closer extracting the meaning of words, however is still very similar to a synonym retriever. Similarly, relation extraction utilizes phrases and parse-trees for determining relationships between entities (again pre-defined entities and relationships), such as Culotta and Sorensen's (2004), Zhang, Zhou, and Aw's (2008), and Sun, Zhang, and Tan's (2011) papers. Here we start to get to the capturing of features, especially contextual or sequential types of features. Others have pursued the use of tree kernels to help with machine learning on text, such as Collins and Duffy's (2001) and Shen, Kruijff, and Klakow's (2005) papers. Each of these papers discuss the use of tree kernels to try to better capture a parse-tree and its dependencies for use in machine learning. This is important with the type of natural language classification we are planning to do, since we hypothesize that the context of the words is just as important, if not more so, than the words themselves.

Since most traditional learning machine learning algorithms rely on feature-based representations of objects, we explored the different types of features that could be used to learn classifications within natural language. Liao (Liao, Alpha, and Dixon 2002) describes features as being local or global. They can be as simple as a single token, a phrase, or something much more complex. Selecting useful and relevant features, as well as deciding how to encode them, can greatly impact the machine learning algorithm's ability to learn good models (Loper and Bird 2002). Therefore, a lot of time is spent on identifying appropriate features, and many people start with everything they can think of. However most of these end up being local representations of the objects (Zelenko, Aone, and Richardella 2003), such as just the words themselves.

Ultimately, we are transforming a document from one set of tokens to another, which is prone to loss of information, such as word sequence. Collobert et al. (2011) discusses common feature categories, such as parts of speech (POS), voice of the sentence, and stemmed root words, while Culotta and Sorensen mention word n-grams, capitalization, and conjunctions (or merging) of many features. Furnkranz (2007) found that using n-grams of length 2 or 3 improved classification over longer n-grams. Forman (2003) suggests the removal of common words (stop words), removal of rare words, and the use of booleans instead of counts for bag of words features. Kernels have also been utilized in place of traditional feature vectors, but were not pursued in our work at this time.

Approach

In order to have a baseline to train against, we took the Electronovision video (Colleran et al. 1964) of the production of Hamlet on Broadway in 1964 and mapped all the movement of the characters for each line of the play-script (Shakespeare 1905). We kept the "sentences" as the way Shakespeare originally divided up his lines of text. Shakespeare nearly always wrote in iambic pentameter (ten syllables per line, with alternating unstressed and stressed syllables) (Mabillard 2012). This meant that a speech like:

> Last night of all,
> When yond same star that's westward from the pole
> Had made his course to illume that part of heaven
> Where now it burns, Marcellus and myself,
> The bell then beating one,– (Shakespeare 1905)

was broken up into five sentences. An alternate approach could have been used where each real sentence was used to determine implied movement or not. This may have helped with the training ratio for movement versus no movement, which will be discussed further in the Experimentation section. However, we chose the phrase-approach because of the frequency of the change in actions being performed within the play. By splitting the sentences to this size, we had a more consistent line-length, were able to more precisely capture a single phrase that might imply movements, and could capture more movements than we could with full sentences.

The main two challenges with mapping this three hour play were in carefully identifying only one movement per line, as well as accurately capturing all the desired movements throughout such a long play, with standardized movement names as seen in Figure 1. Many lines involved multiple movements. To keep things simple, we decided to capture the biggest movement performed by the speaker whenever there were more than one movement for the line.

The key movement types we captured within the Hamlet play can be seen in the list in Figure 2. These movements are for both the speaker and the other characters onstage, and includes how we grouped them for better training capabilities (as will be discussed further in the Experimentation section). As you can see, the majority of movements were captured very few times within the dataset, with the majority being less than 100 instances out of 3477 instances possible.

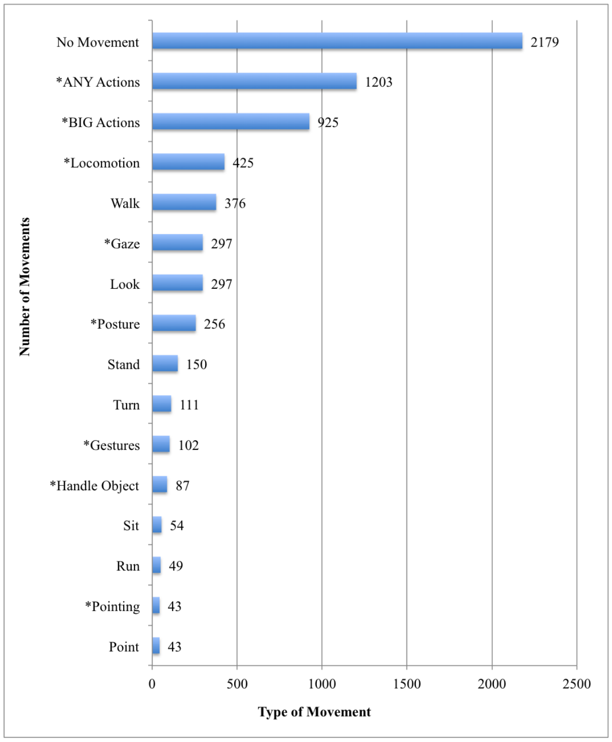


Figure 1: Counts of Distinct Movements Within Hamlet with at Least 40 Instances out of 3477 Lines of Script The Asterisk (∗) Indicates Grouped Categories

Each line of the play's speech was then used to create features for training. We utilized the openNLP package, tied to the Java openNLP implementation, within R to tag each word with its part of speech, along with the RTextTools package (Jurka et al. 2011) for creating our n-grams and bag of words for our text. This information was then chunked into a bag of words approach which used counts of each type of part of speech as a feature. Other features we incorporated into the training included:

- Number of lines for the speaker before this line
- Number of lines for the speaker after this line
- Number of annotations before this line
- Number of annotations after this line
- Number of speech lines since the last movement
- Maximum number of times a word is repeated
- Number of uppercase words in this line of speech
- Count of each punctuation mark within this line

Our hypothesis was that the length of the speech could trigger a movement for the character, such as moving towards the audience due to the start of a monologue. Another assumption was that movements might not occur really close together, to prevent excessive attention and confusion from the audience. Therefore, understanding how long it had been since the last movement was deemed important and a potential aide for learning implied movements. Knowing that there is an annotation coming up (which usually means an

Fighting

- Fighting∗
- Pushing∗

Handle Object

- Hand Object∗
- Pickup Object∗
- Throw Object∗

Change Posture

- Jump∗
- Lie Down∗
- Sit∗
- Stand∗
- Kneel∗

Gaze

Gestures

- Point∗
- Gesture∗
- Nod∗
- Raise Arm∗
- Wave∗

Locomotion

- Walk∗
- Run∗

Other

- Dig∗
- Turn∗
- Climb∗
- Kick∗

Figure 2: Bundled List of Actions Captured within Hamlet The Asterisk (∗) Indicates Items Considered as *Big* Actions

actor will perform some sort of movement), seemed to be useful for determining if a movement should occur now, or would be explicitly provided in the annotation later. Adding the features for punctuation, repeated words, and uppercase words was thought to help with identifying movement that might cause an actor to emphasize what they were saying, such as pointing or gesturing.

We pursued both a part of speech "sentence" and an n-gram bag of words approach for the speech due to Shakespeare's known inclination to make up words and not repeat phrases a lot. We hoped this would help to find patterns in the sequence and frequency of "words," despite being unable to properly turn a parse-tree into a feature vector for training. We are confident that the sequence and dependency tree of the words in conjunction with the words themselves are key in being able to identify implied movement, except with Shakespeare's work due to his jumbling of phrases to fit iambic pentameter. Ideally, also including the number of characters and their positions onstage for each line would be used to help capture the movements related to being upstaged, along with other theatre rule-guided movements.

Experimentation

Once we generated our features for all the lines in the play-script, we fed them into several machine learning algorithms: Maximum Entropy (MaxEnt), Support Vector Machine (SVM), Boosting, and Random Forests (RF). We focused only on the actions the speaker performed during their speech lines, and learning a specific movement or movement type one at a time. Initially, we took a random half of the lines (1739 lines) from the play-script for training the classifiers, and tested on the other half (1738 lines).

However, we found very poor results (same as a random classifier), as can be seen in the Table 1. This was due to having such a large portion of the training set being classified as "no movement," due to often having much worse than a 10:1 ratio of movement to no movement (as can be

	Even Split of Data-Set for Training						2:1 Ratio Split of Data-Set for Training					
Movement	Ratio of Neg:Pos Instances	Recall	Precision	F_1-score	$F_{0.5}$-score	Matthews Correlation Coefficient	Training Size	Recall	Precision	F_1-score	$F_{0.5}$-score	Matthews Correlation Coefficient
Any	1.8:1	0.017	0.455	0.033	0.075	**0.029**	1806	0.917	0.338	0.493	0.386	**0.038**
Big	2.4:1	0.000	0.000	0.000	0.000	0.000	825	0.000	0.000	0.000	0.000	0.000
Gestures	21.4:1	0.000	0.000	0.000	0.000	0.000	153	0.000	0.000	0.000	0.000	0.000
Object	25.1:1	0.000	0.000	0.000	0.000	0.000	134	0.907	0.013	0.026	0.016	**0.007**
Locomot.	5.1:1	0.000	0.000	0.000	0.000	0.000	639	0.953	0.079	0.146	0.097	**0.048**
Gaze	7.3:1	0.000	0.000	0.000	0.000	0.000	447	0.027	0.125	0.044	0.072	**0.037**
Pointing	50.7:1	0.000	0.000	0.000	0.000	0.000	66	0.048	0.026	0.034	0.029	**0.027**
Posture	8.5:1	0.000	0.000	0.000	0.000	0.000	384	0.000	0.000	0.000	0.000	0.000
Fighting	198.1:1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 1: Best Results Per Movement Type While Training on Half of the Data-Set vs a 2:1 Negative:Positive Ratio Data-Set, With All Features, Best Machine Learning Algorithms, and Unigrams. Fighting Did Not Have Enough Instances to Train On. Bolded Ones Performed Better Than Random.

seen in Figure 1). Forman (Forman 2003) discusses the issue of having a substantial class distribution skew (like we see with our Hamlet movement dataset), which worsens as the problem size scales upwards. Most machine learning research does not consider such extreme skews as Forman saw (1:31 on average). Just as we saw with our dataset, we found it very difficult to beat the high accuracy that can be achieved by classifying everything negatively. Forman also mentions that feature selection becomes much more important in these types of situations where the training data is highly skewed.

We first attempted to address this by shrinking down our training set to a more specific set of lines where the ratio of "movement" to "no movement" was closer to a 2:1 ratio, while ensuring we did not use more than half of the annotated movement lines we were trying to classify. This performed marginally better, but still really did not get us past the performance of guessing "no movement" for everything or even a random classification, as can be seen in Table 1.

We also found that we do not have enough examples of detailed movements in Hamlet to be able to classify all movements at a detailed level, such as hand fighting or lying down. Therefore, we were forced to look at the problem more generically than would be useful for actually predicting specific movements. We tried grouping the movements into buckets, as described in the Approach section; however only the posture, gaze, and locomotion came close to a 10:1 ratio, and even learning on those datasets ended up classifying almost everything as "no movement". The main two buckets that could give us almost reasonable results were the ones for any movement and any big movement.

We then looked at the different n-gram approaches to see what would work best to incorporate more of the relationships of the words in the phrases as seen in Table 2. Bigrams appear to have done better than just a plain bag of words (BoW), with trigrams doing slightly worse than the bigrams, but still performing pretty well. 4-grams and 5-grams dropped performance to be closer to unigram performance in most instances. This correlates well with what Furnkranz (2007) mentioned in their work with different n-grams for classifications.

As Forman (2003) discussed, having such skewed training datasets puts more emphasis on the feature sets. Therefore, we pursued several different feature sets and combinations. We began initially with the sentences themselves turned into a BoW of different ngram lengths, along with the other features mentioned in the Approach section.

We then decided to take advantage of Shakespeare's iambic pentameter, which produced the majority of the lines as ten syllables, and a maximum of fifteen words per line. We decided to break these sentences into just the parts of speech (POS) tags as a sentence. This was intended to help with the issue of Shakespeare's writing not including much repetition. With the real sentences broken into BoWs, if we removed sparse words or stop words, we ended up with no words left. However, using the POS tags as sentences, we could get a similar concept, but were able to trim out sparse n-grams.

Finally, we combined the best feature sets described above (in different combinations) to see how it would perform. We chose to use the smaller training set, geared towards a 2:1 ratio of "no movement" to "movement," and focused primarily on classifying any movement within the play. The best classifications were obtained on just the unigrams of the actual speech text, although on average, the part of speech (POS) sentences with the speech sentences as bigrams and the other features did better.

To analyze these statistics, we used the ROCR package within R (Sing et al. 2005) to generate the ROC Curves for the better techniques. We also looked at the overall accuracy, precision, recall, F_1-score, $F_{0.5}$-score, and the Matthews correlation coefficient for each method. We were able to achieve high accuracy, but this was shown to be achievable with just a blind guess of everything to be "no movement". Therefore, the accuracy scores were not useful in determining the goodness of any of our methods.

Looking at precision and recall, we often found we could

Movement Type	n-gram	Machine Learning Algorithm	tp	fn	fp	tn	Accuracy	Recall	Precision	F1-score	F.5-score	Matthews Correlation Coefficient	Direction
Any Mvmt	1	MaxEnt	508	46	997	120	0.376	0.917	0.338	0.493	0.386	**0.038**	
Any Mvmt	2	MaxEnt	505	49	988	129	0.379	0.912	0.338	0.493	0.387	**0.041**	⇑
Any Mvmt	3	MaxEnt	502	52	987	130	0.378	0.906	0.337	0.491	0.386	**0.034**	⇓
Any Mvmt	4	MaxEnt	503	51	987	130	0.379	0.908	0.338	0.492	0.386	**0.037**	⇑
Any Mvmt	5	MaxEnt	501	53	984	133	0.379	0.904	0.337	0.491	0.386	**0.035**	⇓
Locomotion	1	MaxEnt	202	10	2359	267	0.165	0.953	0.079	0.146	0.097	**0.048**	
Locomotion	2	MaxEnt	202	10	2351	275	0.168	0.953	0.079	0.146	0.097	**0.050**	⇑
Locomotion	3	MaxEnt	202	10	2379	247	0.158	0.953	0.078	0.145	0.096	**0.043**	⇓
Locomotion	4	MaxEnt	201	11	2347	279	0.169	0.948	0.079	0.146	0.097	**0.047**	⇑
Locomotion	5	MaxEnt	201	11	2346	280	0.169	0.948	0.079	0.146	0.097	**0.047**	

Table 2: Highlights the Performance of Different N-Grams on Classifying the Different Movement Types on a 2:1 Negative:Positive Ratio Data-Set. Bolded Performed Better Than Random.

Feature Set	n-gram	Machine Learning Algorithm	tp	fn	fp	tn	Accuracy	Recall	Precision	F_1-score	$F_{0.5}$-score	Matthews Correlation Coefficient
Text Only	1	RF	178	376	120	997	0.703	0.321	0.597	0.418	0.510	**0.263**
POS BoW Only	1	RF	158	396	137	980	0.681	0.285	0.536	0.372	0.456	**0.201**
All Features	3	RF	4	550	5	1112	0.668	0.007	0.444	0.014	0.034	**0.018**
POS BoW & Text	3	RF	4	550	5	1112	0.668	0.007	0.444	0.014	0.034	**0.018**
POS BoW & Other	2	Boost	0	554	0	1117	0.668	0	0	0	0	0
POS BoW, Text, Other	1	SVM	0	554	0	1117	0.668	0	0	0	0	0

Table 3: Best Results Per Feature Set, Training on a 2:1 Negative:Positive Ratio Data-Set and Any Movement. Bolded Performed Better Than Random.

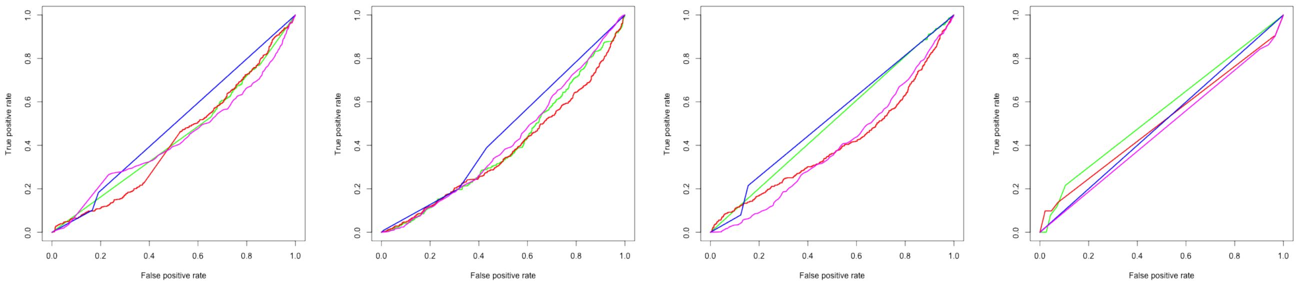
do reasonably well with one, but very poorly with the other. Recall is focused on being able to classify as many positive examples as possible, whereas precision focuses on being more certain of classifying positive examples that really are positive classes. In our case, we are more concerned with making sure that if we identify a line as an implied movement, then there really should be an implied movement with that line. Therefore, precision was more important to us.

Trying to balance these two measures, we looked at the F_1-scores, however this put equal emphasis on both precision and recall. The $F_{0.5}$-score was better since it put more emphasis on the precision than the recall. However, those approaches still left us uncertain to what degree we were able to outperform the random classifier and the guess "no movement" classifier. Therefore, we focused primarily on the Matthews Correlation Coefficient (MCC) measurement, as this takes into account true and false positives and negatives, and is generally regarded as a balanced measure which can be used even if the classes are very skewed like ours. This measure returns a value between -1 and +1. A result of +1 represents a perfect prediction; 0 represents the same as a random classifier; -1 represents 100% incorrect classifications. Using this measure, we found that we were able to do better than the random classifier in many of our tests, as can be seen in the previous tables and in the ROC Curves in Figure 3.

Conclusions

Ultimately, Shakespeare is a more difficult context to use than typical play-scripts due to his tendency to make up words and rephrase things to fit into iambic pentameter. We were able to reasonably tell when some movement should occur, which should at least give us a sanity check for use with our previous work to ensure the characters are moving enough or not. However, the more specific movement types were more difficult to classify due to the limited number of test cases available in Hamlet.

Humans are able to do this with no prior examples, so there must be a way to learn these implied movements. Therefore, future work should include further analysis into tree kernels for machine learning, classifying more detailed movements using additional datasets, and an ability to classify more than one type of movement for a single line. Finally, incorporation of other features may be useful, such as number of characters onstage, locations of all the characters onstage, and other contextual features not included here.



(a) No Text Features
Any Movement
1807 Training Cases
Unigrams

(b) POS BoW Features Only
Any Movements
1807 Training Cases
Unigrams

(c) Text Features Only
Any Movements
1807 Training Cases
Unigrams

(d) All Features
Gesture Movements Only
154 Training Cases
4-grams

Figure 3: ROC Curves Samples for Techniques Utilized;
Red=SVM; Green=Maximum Entropy; Blue=Boosting; Magenta=Random Forests

References

Brooks, A. G. 2012. *Coordinating Human-Robot Communication*. Ph.D. Dissertation, MIT.

Chang, M.-W.; Ratinov, L.; Roth, D.; and Srikumar, V. 2008. Importance of Semantic Representation: Dataless Classification. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, 830–835. AAAI Press.

Chuang, W. T., and Yang, J. 2000. Text Summarization by Sentence Segment Extraction Using Machine Learning Algorithms. *Knowledge Discovery and Data Mining. Current Issues and New Applications*.

Colleran, B.; Gielgud, J.; Shakespeare, W.; Burton, R.; Cronyn, H.; Drake, A.; and Herlie, E. 1964. Hamlet. *Electronovision, Inc.*

Collins, M., and Duffy, N. 2001. Convolution Kernels for Natural Language. In *Advances in Neural Information Processing Systems 14*, 625–632. MIT Press.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research* 12:2493–2537.

Culotta, A., and Sorensen, J. 2004. Dependency Tree Kernels for Relation Extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04. Stroudsburg, PA, USA: Association for Computational Linguistics.

Forman, G. 2003. An Extensive Empirical Study of Feature Selection Metrics for Text Classification.

Furnkranz, J. 1998. A Study Using n-gram Features for Text Categorization.

Jurka, T. P.; Collingwood, L.; Boydstun, A. E.; and Grossman, E. 2011. RTextTools: A Supervised Learning Package for Text Classification.

Kollar, T.; Tellex, S.; Roy, D.; and Roy, N. 2010. Toward Understanding Natural Language Directions. In *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction*, HRI '10, 259–266. Piscataway, NJ, USA: IEEE Press.

Liao, C.; Alpha, S.; and Dixon, P. 2002. Feature Preparation in Text Categorization.

Liu, H., and Singh, P. 2004. Commonsense reasoning in and over natural language. In *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES-2004*, 293–306. Springer.

Loper, E., and Bird, S. 2002. NLTK: The Natural Language Toolkit. *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics-Volume 1* 1:63–70.

Mabillard, A. 2012. Shakespearean Sonnet Basics: Iambic Pentameter and the English Sonnet Style. http://www.shakespeare-online.com/sonnets/sonnetstyle.html.

Platt, J. C.; Cristianini, N.; and Shawe-Taylor, J. 2000. Large Margin DAGs for Multiclass Classification. *Advances in Neural Information Processing Systems* 12(3):547–553.

Schapire, R. E., and Singer, Y. 2000. BoosTexter: A Boosting-Based System for Text Categorization. *Machine Learning* 39(2):135–168.

Shakespeare, W. 1905. *Hamlet*. Horace Howard Furness.

Shen, D.; Kruijff, G. J. M.; and Klakow, D. 2005. Studying Feature Generation from Various Data Representations for Answer Extraction. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP*, 65–72.

Sing, T.; Sander, O.; Beerenwinkel, N.; and Lengauer, T. 2005. ROCR: Visualizing Classifier Performance in R. *Bioinformatics* 21(20):3940–3941.

Sun, J.; Zhang, M.; and Tan, C. L. 2011. Tree Sequence Kernel for Natural Language. In *AAAI Conference on Artificial Intelligence*.

Talbot, C., and Youngblood, G. M. 2012. Spatial Cues in Hamlet. In Nakano, Y.; Neff, M.; Paiva, A.; and Walker, M., eds., *Intelligent Virtual Agents*, 252–259. Springer.

Talbot, C., and Youngblood, G. M. 2013. Shakespearean spatial cues. In *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4-8, 2012 (3 Volumes)*. IFAAMAS.

Wei, Y.; Brunskill, E.; Kollar, T.; and Roy, N. 2009. Where to Go: Interpreting Natural Directions Using Global Inference. *IEEE International Conference on Robotics and Automation* 1–7.

Zelenko, D.; Aone, C.; and Richardella, A. 2003. Kernel Methods for Relation Extraction. *The Journal of Machine Learning Research* 3:1083–1106.

Zhang, M.; Zhou, G. D.; and Aw, A. 2008. Exploring Syntactic Structured Features over Parse Trees for Relation Extraction using Kernel Methods. *Information Processing & Management* 44(2):687–701.