Learning to Gesticulate: Applying Appropriate Animations to Spoken Text

Nate Nichols
Northwestern University
ndnichols@gmail.com

Jiahui Liu
Northwestern University
j-liu2@northwestern.edu

Forrest Sondahl
Northwestern University
forrest@northwestern.edu

ABSTRACT

We propose a machine-learning system that learns to choose amongst human-like gestures to accompany novel text. The system is trained on scripts (which are comprised of speech and animations) that were hand-coded by professional animators and shipped in games built on top of the Source game engine. The system first extracts features from the text that was spoken, and maps these features to the gestures that accompany the speech. We have experimented with using a number of features of the text, including n-grams of the words themselves, emotional valence of the speech, and part-of-speech tagging. Using naïve Bayes classifiers, the system learns to associate these features with appropriate gestures. Once trained, our system can be given novel text to which it will attempt to assign appropriate gestures. We examine the accuracy of the system by using n-fold cross-validation techniques over our training data, as well as a user study, composed of subjective evaluation of the results. In the user study in particular, our system was able to outperform random application of gestures. Although there are many possible applications of automated gesture assignment, in particular we hope to apply this technique to the problem of coordinating human-like gestures to the text spoken by avatars in an automated news show.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *text analysis.*

General Terms

Human Factors

Keywords

Machine Learning, Naïve Bayes, Gestures, Animation

1. INTRODUCTION

When humans converse, they speak not only with their voices, but with their whole bodies [3]. Pick several people randomly off the street, and closely observe their hands or their eyebrows while they talk. You will find that there is a constant flow of movement that is coordinated with their speech. Although it is not obvious how each gesture relates to the individual words they are speaking, or more generally to the content being discussed, the overall effect is natural. So natural, in fact, that unless you are purposefully thinking about it, you are unlikely to notice how many gestures occur. Conversely, if you were to watch a computer avatar give the same speech with her hands resting at her sides and her face placid, it looks fake, even if the face and skin have been photo-realistically rendered. In short, gestures are a critical component for the creation of life-like animation. For this reason, video game and movie maker pay teams of professional animators to choose appropriate animations for their virtual actors to perform while they speak their script.

However, if the script is not known ahead of time, hiring professionals to hand-code gestures is not an option. Such a case requires a method for automatically choosing appropriate gestures to match the text that is being spoken. More comprehensively, these gestures should match the mood, temperament, situation, and cultural norms of the avatar. However, in this paper we limit the scope of the problem to predicting appropriate gestures solely on the basis of the text (with an implicit presumption that further mechanisms could be added to make adjustments for these additional factors).

Our particular goal is to apply this system to an automated news show, *News at Seven* [7], wherein virtual newscasters (avatars) read current news stories accompanied by photos and video footage. We hope to improve the realism of this virtual news show through richer and more applicable gestures for the avatars. To accomplish this, we take advantage of the very work that we were seeking to avoid, i.e. hand-coded animation scripts that accompany video games. In particular, we utilize the scripts from several titles written using the Source game engine, which is the same engine used by *News at Seven*. These scripts contain text that has already been annotated with gesture animations by professional animators, and this comprises our training corpus. We then apply a naïve Bayesian machine learning algorithm to correlate features of the text with the gestures that were chosen. Using this correlation information, given a novel piece of text (such as a news story or opinion piece), the classifiers will choose gestures that are appropriate to accompany the text.

2. RELATED WORK

Although we are not alone in seeking a system that can automatically apply appropriate animations to arbitrary text, we believe that our particular approach to be a novel one. Some previous approaches utilize specific hand-coded rules to determine gestures. Cassell et. al. have designed a toolkit (BEAT) that is based on rules that were derived by extensive research into human conversational behavior [2]. BEAT is extensible so that new rules can be added, and it also allows animators to make

EECS 395-22 Machine Learning, 2007, Evanston, IL, USA.

manual changes to the animation choices after the initial first pass, although this would not be a possibility in a fully automated environment. Also, the rules can be made somewhat more general through the use of parts-of-speech analysis, and embedded word ontology modules (e.g. WordNet [6]). Although a carefully constructed rule based approach could yield good results, the expertise required to construct good rules could be an obstacle in many cases. Also, it is not at all clear that such a setup scales; much of its efficacy is based on a semantic understanding of the text, a notoriously difficult problem to solve. There has also been some previous work that attempted to integrate a basic rule-based approach with a model of an actor's internal emotional state [4]. In contrast, our actors do not have an internal model of emotion, but instead we are able to extract some emotional information from the text, using a sentiment classification system developed by Owsley et. al. [8], and use this as another factor when determining an appropriate gesture to perform. In general these rule-based systems depend heavily on the knowledge of the rule designers and their ability to explicitly model text-to-gesture mapping. Conversely, our machine-learning approach allows such rules to be defined implicitly, through examples. Our system is the first machine-learning/corpus-based approach to automated gesturing, and leverages the collections of existing examples created by professional animators. Although the current available training data is not as much as we'd like, there exists enough already to generate interesting, useful results. Furthermore, as computer animation becomes more and more prevalent it seems clear that the amount of training data available will only increase, further improving our system.

It should be noted that our system is not intended to capture and model all of human gesticulating. Because our system strictly uses features of the text, it will never be able to understand "Bob went that way" and point at the way Bob went, for instance. It is capable of learning associations between the word "big" and spreading your arms far apart, or other so-called iconic gestures, however [1]. We settled on this compromise for three reasons. The first is that it is clear that the problem becomes disproportionately more difficult when trying to accommodate the whole range of human gestures. Secondly, although large, iconic gestures are often what first spring to mind when imagining "gestures", small "beat gestures", like hand-waves, head-tilts, and body-leans, actually make up the bulk of human gesturing. Finally, due to graphical engine constraints, animators are often in a position where gestures are restricted to be chosen from a finite set of animations, which means that the "perfect" human-like gesture is probably unavailable. Rather than prompt despair, these facts merely emphasize the necessity for setting more modest goals. We are not trying to teach our avatars to gesture precisely like humans; rather we are trying to get them to gesture enough to not seem stiff, but not too much that they perform obviously fake, inappropriate animations.

3. METHOD

The core idea of our project is using hand-scripted scenes from modern video games to learn correlations between features of speech and appropriate gestures using a naïve-Bayes classifier. In our work, the spoken texts in scripts are considered to be the documents, and the co-occurring animations are the possible classifications.

The scripts we used came from the popular game *Half-Life 2*. This game was chosen because it has a fairly large number of scenes, and code already existed to parse the scenes. A scene is typically composed of a few spoken sentences and gestures that were applied and chosen by the game designers. These speech and gesture events are stored in a "timeline" format, such as one might see in video-editing software.

Because of this format, we are able to adopt the timing information in the original scripts to determine the co-occurrence of speech and gestures. We split the text of a scene, into serial separate "chunks" that co-occur with zero gestures, one gesture, or more than one gesture. If one gesture is being performed when a "chunk" is spoken, then that chunk is classified as an instance of that gesture. If no gesture is being performed, then the chunk is classified as an instance of the special "NONE" animation. Finally, if a chunk co-occurs with more than one animations (which is typically a smaller "accent gesture" layered on top of a larger gesture), then that chunk is classified a *combination gesture*. The following features are extracted from the chunks and used in classification:

1. N-Grams: Unigram, bigram, or n-grams (n>2) are used to determine the relationship between the content of the texts and the corresponding gestures.
2. Emotional valence: To capture the relationship between the emotional state of the avatars and the gestures they use, we used the sentiment classification web service to get the emotional valence for each word in the texts. Because our naïve-Bayes classifier only takes discrete features, we encode the emotional valence to integer values using a simple equal-interval binning method.
3. Part-of-speech (POS): The texts are tagged with the POS tagger developed by Liu [5]. The POS tags for the words are used as syntactic feature for the classifier.

One potential weakness of the system is the relative poverty of the vocabulary used (only about 1,700 unique words); not only that, but the vocabulary has a strong "videogame" bent (the word "zombie" occurs five different times). Because all words have a POS, and the emotional valence system is drawn from a much larger corpus, the emotional valence and POS classifiers will help the system extend to novel domains. In addition, to ensure the system did not overly train on specific names, we automatically replaced the proper names in the text with the token "*__Proper_Name__*". Therefore, the classification of the texts is not influenced by any specific names in the speeches.

To classify new scripts, the system reads the text of each speech event, and feed that text through our trained classifier. The system then chooses the most likely gesture (which may be a "combination gesture") and creates a new gesture event for that gesture that begins at the same time of the speech event. Because the prior probability of "NONE" is much higher than any other gestures, the system ignores "NONE" in the results of classifications. Rather, if the probability of the top gesture is below certain threshold, the system assigns "NONE" to the speech event.

4. EXPERIMENT

4.1 Objective Performance of Classifiers

From the *Half-Life 2* game engine, we extracted 1,037 scripts with speech events. Within the scripts, there are 234 distinct gestures. Many of the gestures co-occur with other gestures, yielding 2,707 gestures including the "combined gestures."

To evaluate our method, we used 10-fold cross validation on the 1,037 scripts. For the training data, we labeled the spoken texts with the co-occurring gestures as described in Section 3. Various naïve-Bayes classifiers using different features were then trained on the training samples. We also tried some combination of the classifiers using weighted sum of the probabilities of the classifiers. For each test script, the text of speech event in the script is fed into the classifier. The classifier assigns a gesture, which may be a "combined gesture", to the speech event. The system compares the assigned gesture with the gestures manually designed by the game animators in the original scripts. If the gesture suggested by the classifier is a subset of the original gestures, the test script is counted as a success. (Being a subset was counted as correct because we know the system chose an animation that the professionals chose, even if it didn't choose *all* the animations the animators chose.) The system summarizes the correctness of the classifier on all the test scripts and computed the accuracy of the classifier. Table 1 reports the performance of the various classifiers on the corpus.

Classifier	Performance
Classifier with 1 feature: Unigram	18.5%
Classifier with 1 feature: Bigram	22.9%
Classifier with 1 feature: 3-Gram	24.7%
Classifier with 2 features: Emotional valence and POS	16.5%
Combination of classifier 0.4*Unigram + 0.6*Bigram + Trigram	23.0%
Combination of classifier 0.4*Unigram + 0.6*Bigram + Trigram+ 0.4 * Emotional valence and POS	19.8%
Applying most common animation	24.0%

Table 1. Performance of Classifiers

The main thing these results show is that this performance metric is not a particularly useful one. Applying the most common animation to every single speech instance yielded almost the highest score at 24%. (The most common animation happens to be a small body-lean-forward accent that is apparently used in a quarter of all gestures.) Of course, to a human observer, the same small body tilt on every single spoken line looks ridiculous. On the other side, semantically identical gestures may be scored as misses by the test. If the original animators applied a tilt-head-right gesture, and our classifier applied a tilt-head-right-2 gesture,, it would be marked as a miss, even though the animations are essentially identical. Because we are ultimately interested in gesturing that looks appropriate, not slavish recreation of the work of professional animators, we performed a user study to see how real humans evaluated the various animations.

4.2 User Study

First, we chose ten scenes from our corpus at random; we then recorded these scenes three different times. The first time, we used just the animations chosen by the game designers, and the second time, we used the animations chosen by our classifier trained on uni- and bi-grams. The third time, we randomly assigned gestures to the scene, based on how often that gesture occurred over all the scenes in the corpus. We consider this random application to be the simplest thing that could possibly work; it doesn't have the obvious repetition of choosing the same common application over and over, but also doesn't have any intelligence behind it at all. We then randomized the order of the three different takes of each scene, and recorded a video.

Twenty human volunteers were asked to watch the video (comprised of three different takes of ten scenes) and rank each scene relatively by how appropriate the animations were. So, if for scene one a participant thought the second take was best, followed by the first take, followed by the third, they would record "213" for that scene. Thus, we had 200 separate ranked scenes (20 volunteers ranking ten scenes a piece.)

Not surprisingly, the original animations were chosen to be the best: 62% of the time; the classifier's animations were chosen 20% of the time; and the random animations were preferred 17% of the time. The classifier was also more likely than the randomized to be chosen as second best, being picked 41% of the time compared to only 34% for the random animations. The average chosen ranking for the original was 1.5, for the classifier was 2.19, and for random was 2.31. The classifier scored higher than random 56% of the time, and higher than the original animations 25% of the time.

Perhaps the most interesting result here is how relatively poorly the original animations fared. These animations were chosen by professional animators and could be applied anywhere in the scene; our gestures were chosen by a classifier trained on a very small corpus, and can (currently) only be applied at the beginning of a speech event. Overall, this is a very encouraging result; it strongly implies that there isn't a single "gold-standard" gesture that looks appropriate at a certain situation; rather, there seem to be a large number of animations of varying "rightness." We think this bodes well for future automatic-gesturing systems. Furthermore, the fact that the classifier outscored the random animations encourages us further work on a naive-Bayes approach may continue to be fruitful.

We believe the principle issue facing the classifiers is the scarcity of the training data. On average, there are 2.3 training text documents for each gesture. Most of the "combined gestures" only occur once and are rarely applied during testing because of their low prior probabilities. For the 234 generic gestures, there are 10.2 training documents on average. All of the training documents are very short, usually with less than 10 words. Compared with the whole vocabulary of 1,714 words, there are simply not enough instances of each word.

5. FUTURE WORK

An obvious plan is to collapse the animations into semantically-distinct bins. For testing and training purposes, for example "big-shrug-1", "big-shrug-2", and "big-shrug-left" would all be considered the same gesture. This has a couple of helpful effects. Firstly, it would give us more instance of each gesture classification and fewer possible classifications in general. Secondly, it would make our automated testing more like the user study testing; this means we can revise and update the system, testing regularly automatically, and have a better idea of how well the system will fare when view by a real public. Finally, it will help us expand our training corpus by allowing us to use the scripts for more games. The animation sets are distinct between games, and we can only apply animations that work within *Half-Life* 2. Therefore, training on other games confers no benefit, because we couldn't apply any of those animations anyway. If we had pan-games semantic bins, however, we could always choose an equivalent *Half-Life* 2 animation and use the other games scripts to successfully expand our corpus.

We also intend to experiment further with improving the classifiers, and combining them for a final gesture recommendation. We also hope to use a similar technique to learn facial expressions that are also present in the scripts.

6. CONCLUSION

We use a machine learning/corpus-based technique to automatically suggest gestures according to the text spoken by avatars. We intended to use this technique for the automated gesticulation of an automated news show, but we believe this same general technique holds promise for other situations as well. In fact, returning to the source from whence it came, there could be great demand for a system such as this in the video game industry. By functioning as an animation first-pass for the game designers, we will be able to automate the basic animation required to look human-like, and free the animators to work on more interesting and complicated animations. A similar system could also be used to make characters in online games like *World of Warcraft* appear more dynamic and convincing. We look forward to continuing development on the system and seeing where it leads in the future.

7. REFERENCES

[1] Beattie, G., Shovelton, H. Mapping the Range of Information Contained in the Iconic Hand Gestures that Accompany Spontaneous Speech. *Journal of Language and Social Psychology, 18, 4* (1999), 48-462.

[2] Cassell, J., Vilhjalmsson, H., Bickmore, T. BEAT: the Behavior Expression Animation Toolkit. *ACM SIGGRAPH*, 12-17 August 2001.

[3] Cassell, J., Vilhjalmsson, H. Fully Embodied Conversational Avatars: Making Communicative Behaviors. *Autonomous Agents and Multi-Agent Systems*, v.2 n.1, p.45-64, March 1999.

[4] Gratch, J., Marsella, S. Tears and Fears: Modeling emotions and emotional behaviors in synthetic agents. *Proceedings of the fifth international conference on Autonomous agents, (2001),* 278-285.

[5] Liu, H. MontyLingua: An end-to-end natural language processor with common sense. Available at: web.media.mit.edu/~hugo/montylingua, 2004

[6] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K., Introduction to Wordnet: An On-line Lexical Database, 1993.

[7] Nichols, N., Owsley, S., Sood, S., Hammond, K. *News at Seven*: An Automatically Generated News Show. Submitted to *WWW2007*, May 8-12, 2007.

[8] Owsley, S., Sood, S., Hammond, K. Domain Specific Affective Classification of Documents. In *Proceedings of the AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*., March 2006.