

Learning to Gesture: Applying Appropriate Animations to Spoken Text

Nathan Nichols, Jiahui Liu, Bryan Pardo, Kristian Hammond, Larry Birnbaum

Northwestern University

2133 Sheridan Road

Evanston, IL 60208

{ndnichols, j-liu, pardo, hammond, birnbaum}@cs.northwestern.edu

ABSTRACT

We propose a machine learning system that learns to choose human gestures to accompany novel text. The system is trained on scripts comprised of speech and animations that were hand-coded by professional animators and shipped in video games. We treat this as a text-classification problem, classifying speech as corresponding with specific classes of gestures. We have built and tested two separate classifiers. The first is trained simply on the frequencies of different animations in the corpus. The second extracts text features from each script, and maps these features to the gestures that accompany the script. We have experimented with using a number of features of the text, including n-grams, emotional valence of the text, and parts-of-speech. Using a naïve Bayes classifier, the system learns to associate these features with appropriate classes of gestures. Once trained, the system can be given novel text for which it will attempt to assign appropriate gestures. We examine the performance of the two classifiers by using n-fold cross-validation over our training data, as well as two user studies of subjective evaluation of the results. Although there are many possible applications of automated gesture assignment, we hope to apply this technique to a system that produces 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 we converse, we communicate not only with our voices, but with our entire bodies [3]. Although hand movements or eyebrow-raises are often not consciously noticed by the listener, an *absence* of these human-like movements certainly is. The problem only becomes more pronounced as computer graphics

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. MM'07, September 23-29, 2007, Augsburg, Germany. Copyright 2007 ACM 1-59593-447-2/06/0010...\$5.00.

improve. The closer an avatar looks to a real human, the more we expect the avatar to also act in a human way, gestures and all. For this reason, video game and movie producers pay professional animators to choose appropriate animations for their virtual actors to perform as they act out a script.

Unfortunately, professional animators are expensive; furthermore, if the script is not known ahead of time, hiring professionals is not an option. Such a situation requires a method for automatically selecting appropriate gestures to match the text that is being spoken. In this paper, we propose a novel method for automatically assigning gestures to accompany texts. Our method uses existing scripts comprised of speech and animations as training data and learns to assign gestures to novel text. A more comprehensive system would account for the mood, temperament, situation, and cultural norms of the avatar. In this paper we limit the scope of the problem to predicting appropriate gestures solely on the basis of the text (with the presumption that further mechanisms could be added to make adjustments for these additional factors).



Figure 1. An avatar speaks and gestures

Our ultimate goal is to apply this system to an automated news show, *News at Seven* [7], wherein virtual newscasters (avatars) present current news stories accompanied by photos and video footage. We hope to improve the realism of this virtual news show through richer gestures for the avatars. To accomplish this, we take advantage of the result of the very process that we are

seeking to avoid, i.e. hand-coded animation scripts that accompany video games. In particular, we utilize the scripts from several games written using the Source game engine, which is the 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 have tried two different approaches to applying animations to novel text. The first, referred to as the “stochastic classifier,” is simply randomly choosing amongst the animations, with each animation weighted by how often it occurred in the training data. For the second approach, we trained a naïve Bayesian machine learner to correlate features of the text with the gestures that were chosen by the professional animators. Both approaches and their performance are discussed below.

2. RELATED WORK

Although we are not alone in seeking a system that can automatically apply appropriate animations to arbitrary text, we believe our machine learning approach to be a novel one. Cassell et. al. have designed a toolkit (BEAT) that is based on rules that were derived by observing human conversational behavior [2]. BEAT is extensible so that new rules can be added, and the rules can be made 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 theoretically yields good results, the expertise required to construct good rules is difficult to find. Also, it isn't clear that such a fragile setup scales well; 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]. We use this information when determining an appropriate gesture to perform. In general, 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. We believe that 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 nearly as extensive as we would like, there exists enough already to generate interesting, useful results. Furthermore, as computer animation becomes more and more prevalent, it is clear that the amount of training data available will only increase, further improving our system.

3. METHOD

The stochastic classifier is trained strictly on the number of occurrences of each specific animation in the training corpus. There is a large variance in how often each animation actually occurs in the game. Some animations are intended for specific points of the game (climbing a certain ladder for example), while others are intended to be used throughout (like shrugging shoulders or tilting the avatar's head.) The system records how often each animation was used throughout the corpus so that we

can weight the animations appropriately in our simpler stochastic system.

More work was required to prepare the data for the naïve Bayes classifier. Because of the paucity of training data available, we collapsed specific animations into “animation classes.” For example, the character Alyx may have an animation *alyx_gesture_13* where she shrugs her shoulders, and the character Barney may have an animation *barney_animation_42* where he makes the exact same gesture. If our classifier classified strictly according to specific animations (like *alyx_animation_13*), then it may “split the vote” between the two shrugging animations, and neither would be chosen. In order to give the classifier more examples of each target classification, the animations were split by hand into 17 different groups or classes of animations (like “both hands up” or “shrug”), agreed upon by two independent researchers. There was approximately 75% initial agreement between the two researchers, who then reclassified the disputed animations until both were satisfied.

The scripts we used for training and testing came from the popular game *Half-Life 2*. Each script is a machine-readable description of a scene, and each scene contains one or more pieces of speech and corresponding animations chosen by the game designers. These text and gesture events are stored in a “timeline” format, such as one might see in video-editing software. *Half-Life 2* was chosen because compared to other video games it has a fairly large number of scripts (which are stored as simple text files).

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 as a *combination gesture*. The following features are extracted from the text chunks and used in classification:

1. N-Grams: Unigram, bigram, or trigrams are used to determine the relationship between the content of the texts and the corresponding gestures.
2. Emotional valence: We automatically tag each word in the text with a floating point positive/negative valence score using Owsley's sentiment classification system.
3. Part-of-speech (POS): The text is tagged with the POS tagger developed by Liu [5]. The POS tags for the words are used as syntactic features for the classifier.

Because we would like to use the system to apply animations in non-*Half Life 2* domains, one weakness of the naïve Bayes classifier approach is the small size of the text vocabulary available (only about 1,700 unique words); besides being small, the vocabulary also has a strong “videogame” slant (the word “zombie” occurs five different times). Because of the small training vocabulary, it seems likely that the system will never have seen many of the words in a news story, for instance, and so would not know where to place animations. Because all words

have a part-of-speech, and the emotional valence system is drawn from a much larger corpus, the emotional valence and part-of-speech classifiers will help the system extend to other domains. In addition, to ensure that the system not associate gestures to proper names, we automatically replaced the proper names in the text with a special token, ensuring that the classification of the text is not influenced by any specific names in the speech. Because the stochastic classifier works independently of the text, it would function as well on totally new words as it would on the scripts it trained on.

To classify new scripts, we make the simplifying assumption that each speech event has an associated gesture event that occurs at the same time. The system reads the text of each speech event, and feeds that text through one of the two classifiers. If the naïve Bayes classifier is used, the system assigns the most likely animation *class* to the text. Once a class of animations is chosen, a specific animation is chosen by weighting each animation in the class by how frequently it was used in the training corpus, and then stochastically choosing amongst the weighted values. The stochastic classifier works similarly—choosing stochastically amongst weighted values—but considers all animations in the training corpus, not just animations from a specific class.

4. EXPERIMENT

4.1 Experimental setup

From the *Half-Life 2* game engine, we extracted 1,037 scripts with speech events. Within the scripts, there are 234 distinct gestures, which we collapsed down to 17 different classes of animations. 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 feature sets were trained on the training samples. We also experimented with combining classifiers using a weighted sum of the probabilities of the classifiers (Table 1). For each test script, the text of each 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(s) with the gestures manually designed by the game animators in the original scripts. If the class of the gesture suggested by the classifier is a subset of the original gesture classes, 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 did not choose *all* the animations the animators chose.

4.2 Experimental results

An interesting result is that simply applying the most common animation class (“lean forward”) to every scene was judged as 37% percent accurate. (Apparently, a little over a third of all speech events in the game co-occur with some member of the “lean forward” animation class.) Our more complicated classifier approaches performed just a little better according to the automated testing. It seems likely, however, that while simply applying the same animation class to every single speech event may score fine on automated testing, it would ultimately look stale

and ridiculous to human viewers. Because we are ultimately interested in gesturing that looks appropriate, we performed two user studies to see how real humans evaluated the various animations.

Table 1. Performance of class-of-animations classifiers

Classifier	Performance
Applying most common animation class	37.3%
NB Classifier using text unigrams	37.0%
NB Classifier using text bigrams	37.3%
NB Classifier using text trigrams	38.5%
NB Classifier using emotional valence	33.5%
Weighted sum of NB classifiers 0.4 * unigrams + 0.6 * bigrams + 3-grams + .04 * emotional valence and POS	45.0%

4.3 User study setup

We performed two similar but distinct user studies in order to test the efficacy of our system. For the first test, we chose ten scripts from our corpus at random. Then, we created animated gestures for each scene using three different methods. The first method was to use the animations chosen by the game designers. The second method used the animations chosen by our naïve Bayes classifier trained on unigrams and bigrams of the text in a training set. The third method used the stochastic classifier to assign gestures based on the prior probability of occurrence for each gesture when calculated over the full set of scenes in the corpus. We then randomized the order of the three different takes for each of ten scenes, and recorded a video of all thirty scenes.

Twenty human volunteers were asked to watch the video and rank each set of three takes (three different sets of gestures) by how well the gestures in each take corresponded to the text. For example, given a scene with three takes, a participant would enter “213” if the gestures in the second take best corresponded to the text, the gestures in the first take were less appropriate and the gestures in the third take were least appropriate. Collecting the data gathered from the users resulted in twenty ranked orderings for each of the ten scenes.

4.4 User study results and discussion

Not surprisingly, the original animations were chosen to be the best overall. The original animations were chosen as best 62% of the time; the naïve Bayes classifier's animations were chosen 20% of the time; and the stochastic classifier's animations were preferred 17% of the time. The naïve Bayes classifier's animations was also slightly more likely than the stochastic classifier's to be chosen as second best, being picked 41% of the time compared to 34% for the stochastic animations. The average chosen ranking for the original was 1.5, for the naïve Bayes classifier was 2.19, and for stochastic was 2.31. The naïve Bayes classifier performed better than the stochastic selection 56% of the time, and higher than the original animations 25% of the time.

Because we are only interested in generating plausible animations, not besting professional animators, we designed a second user study. Again, we took three different takes of each of ten scenes. One was the originally chosen animations, one had animations chosen by the stochastic classifier, and one had animations chosen by the naïve Bayes classifier, again trained on unigrams and bigrams of the text. For this study, twelve participants (with some overlap with our first study) rated each scene independently from all the other scenes. For each of the thirty scenes, the participant simply noted whether or not a human reading the text might plausibly make the same gesture. In this study, the original animations were chosen as plausible 86.3% of the time, the stochastic classifier's animations were chosen as plausible 59.1% of the time, and the Naïve Bayes classifier's animations were plausible 52.7% of the time.

There are a number of conclusions to draw from these results. The first is that human animators are not perfect; their hand-chosen animations were chosen as implausible almost 15% of the time in our second study.

Secondly, there doesn't seem to be a "gold-standard" animation that looks correct while all gestures others look incorrect. In our second study, although the classifiers almost always chose different animations, these differing animations were frequently both labeled plausible. It seems, then, that there is a landscape of gestures that have varying levels of "rightness" or plausibility.

Thirdly, the more complicated naïve Bayes classifier had very similar performance to the simpler stochastic classifier. We believe the principle issue facing the naïve Bayes classifier is the scarcity of the training data, although a more advanced classifier (such as a support vector machine) may perform better on the small corpus. Many of the animations occur infrequently (there are only an average of ten instances of each of the 234 specific gestures) and the training documents have little text (usually less than 10 words.) Although collapsing the gestures to animation classes did help to increase the number of examples of each possible classification, there were still very few examples. We believe that the naïve Bayes classifier would perform better than the stochastic given more training data.

5. FUTURE WORK AND CONCLUSION

The most obvious way to improve the system is by growing the size of the training corpus. Although a number of games built on the Source engine share the same script structure, the animation sets are distinct between games, and we can only apply animations that work within *Half-Life 2 to News at Seven*. If we collapsed similar animations across games into the same animation classes, however, we could always choose an equivalent *Half-Life 2* animation and use the other games scripts to successfully expand our corpus. We would also like to conduct a more in-depth user study, comparing our methods to other gesture application systems, like BEAT.

It should be noted that our system is not intended to capture and model all of human gesturing. 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. First, 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, and our system is able to apply these. 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 for an unseen piece of text is likely unavailable. We are not trying to teach our avatars to gesture precisely like humans; we are only trying to get them to gesture appropriately enough to not seem stiff and robotic.

We use a machine learning/corpus-based technique to automatically suggest gestures according to the text spoken by avatars. We intend to use this technique for gesturing in 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 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. <http://www.newsatseven.com>
- [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