Classifying Turn-Level Uncertainty Using Word-Level Prosody

Diane Litman[1], *Mihai Rotaru*[2], *Greg Nicholas*[3]

[1]Computer Science Department and LRDC, University of Pittsburgh, Pittsburgh, PA, USA
[2]Textkernel B. V., Amsterdam, The Netherlands
[3]Computer Science Department, Brown University, Providence, RI, USA
litman@cs.pitt.edu, mich.rotaru@gmail.com, gdn@cs.brown.edu

Abstract

Spoken dialogue researchers often use supervised machine learning to classify turn-level user affect from a set of turn-level features. The utility of sub-turn features has been less explored, due to the complications introduced by associating a variable number of sub-turn units with a single turn-level classification. We present and evaluate several voting methods for using word-level pitch and energy features to classify turn-level user uncertainty in spoken dialogue data. Our results show that when linguistic knowledge regarding prosody and word position is introduced into a word-level voting model, classification accuracy is significantly improved compared to the use of both turn-level and uninformed word-level models.
Index Terms: emotion recognition, speech dialogue systems

1. Introduction

Affective computing investigates systems that detect, adapt to and/or express emotions.[1] Here we focus on *emotion detection for spoken dialogue systems*, where much prior research has examined the utility of different feature types (e.g. prosodic [2], lexical [3], and contextual [4]) for classifying emotion using supervised machine learning. Since the majority of this work has annotated emotions in spoken dialogue data at the user turn level[2], most classification work has also computed features at the turn level. However, recent work suggests that using acoustic-prosodic features at sub-turn levels can improve classification accuracy, e.g. [6, 7]. The intuition behind using sub-turn features is that they offer a better approximation of the acoustic-prosodic profile, and that emotion is often not expressed over entire turns. However, sub-turn features introduce at least two complications for classifying turn-level emotions: the choice of sub-turn granularity, and the many-to-one relationship between sub-turn units and the turn-level emotion.

Since our study is in the context of a spoken dialogue system, we use words as our sub-turn granularity. Word segmentation is available for free as a byproduct of speech recognition, and the terse nature of user turns in typical human-computer dialogues makes coarser granularity less practical (e.g. turns end up with only one sub-turn segment). The more central problem to our study is the many-to-one relationship between multiple sub-turn units and a single turn-level emotion. One popular approach has been to first use supervised learning to predict an emotion for each individual sub-turn unit, then combine these predictions to derive the turn-level emotion [8, 6, 9]. In our own prior work [6, 10], we used word-level pitch features to predict word-level emotions, then used majority voting to derive the turn-level emotion. Here we show how to further improve a voting-based approach, by designing and evaluating voting methods that are informed by linguistic knowledge regarding word position and prosodic manifestations of uncertainty.

[1]We use the terms "emotion" and "affect" loosely, as speech researchers have found that the narrow sense excludes states where emotion is present but not full-blown, including arousal and attitude [1].

[2]The word-level annotation of [5] is a notable exception.

ITSPOKE$_{25}$: What relations do the directions of acceleration and net force have to each other?

Student$_{25}$: they are equal (ASR: they are equal) *[nonUncertain]*

ITSPOKE$_{26}$: Very good. Again, in this relationship, what is the duration that a force acts on a body and the duration of the body's acceleration?

Student$_{26}$: as long as they're in contact (ASR: as long as thank contact) *[Uncertain]*

Figure 1: *Dialogue excerpt - human transcript, speech recognition hypothesis (ASR) and human uncertainty annotation.*

2. Corpus and annotation

Our corpus consists of 9588 user turns from 347 spoken dialogues between 80 users and ITSPOKE (**I**ntelligent **T**utoring **SPOKE**n dialogue system) [11], a speech-enabled version of the WHY2-ATLAS conceptual physics tutoring system [12]. ITSPOKE first analyzes a user essay response to a physics problem for mistakes and omissions, then engages in a spoken dialogue to remediate the identified problems. As shown in Figure 1, the dialogues follow a "tutor question - user answer - tutor response" format, which is hand-authored beforehand in a hierarchical structure. User turns in our dialogues are relatively short, with 2.8 words per turn on average.

All 9588 user turns in our corpus have been annotated[3] on an *Uncertain–nonUncertain* dimension. Our research has focused on uncertainty due to its frequency in our data, and its important role in tutoring dialogue, e.g. [13]. The majority of turns are labeled as *nonUncertain* (78.70%).

3. Features

While previous work has used a variety of information sources for emotion prediction, e.g. [8, 2, 3, 4], we focus only on *pitch* and *energy*. Both correlate with and are predictive of various emotions, e.g. [2, 3, 8, 14, 15], and our intuition behind word-level features (i.e. that they offer a better contour approxima-

[3]A second annotator annotated 4,895 turns, with 0.74 Kappa.

tion and that emotion is not necessarily expressed over an entire turn) translates well to energy and pitch.

To extract the pitch (f0) and energy (RMS) information, we used the Entropic Research Laboratory's pitch tracker, *get_f0*, with no post-correction (www.speech.kth.se/software/#esps). From each of the two contours (pitch and energy) we extract 10 features. The minimum, maximum, mean, and standard deviation of the contour values are commonly used for emotion detection [3]. Inspired by [8], we also use the following features that offer a better approximation of the pitch contour: the first value, the last value, and linear regression coefficient and error. Finally, we use the second order and the zero order coefficients of the quadratic interpolation, which relate to the Tilt model [16].

To compute turn-level features we use the pitch/energy contour over the entire turn, and thus have 20 turn-level features (10 from pitch and 10 from energy). To compute word-level features, for each word we use only the contour data points that lie within the word boundaries. We also include two positional features (distance in words from the turn beginning and turn end) to treat the words in a turn as a sequence not as a "bag-of-words". Even though the word-level feature set is larger than the turn-level one due to positional features, without the two positional features the word-level approach will be deprived of the order information implicit in the turn-level approach.

The computation of word-level features requires a segmentation of turns into words, which we automatically obtained from human turn transcriptions using the Sphinx2 recognizer in forced alignment mode; alignment using noisier speech recognition output is left as future work.

4. Predicting uncertainty

For our experiments we use Weka's (www.cs.waikato.ac.nz/ml) "AdaBoost" classifier to boost a "J48" decision tree learner, which has produced robust results in our prior work [6]. To evaluate performance we run 10 trials of 10-fold cross-validation, and compare models using paired two-tailed t-tests.

4.1. Baselines: turn-level and word-level majority voting

Predicting uncertainty using a *turn-level model* is straightforward and uses a standard machine learning approach. During training, the learner is given 20 turn-level features and the corresponding turn-level class. During testing, the resulting classifier is presented with turn-level features extracted from a testing turn and predicts the class for that turn.

At the word-level, the many-to-one relationship between the multiple words in the turn and the single turn-level uncertainty label prevents us from using this same approach. As a baseline *word-level model*, we use the following **majority** vote model from our previous work [6, 10]. In the training phase, each word from a training turn is labeled with the turn class to produce a training instance: its word-level feature and the label. The machine learning classifier is trained on this data to produce a *word-level classifier* (i.e. we predict word labels). For each testing turn, we then predict the class of each word in the turn using the word-level classifier trained in the first phase. To produce the turn class, we combine the word classes using majority voting (ties broken randomly).

Table 1 shows accuracy, precision, recall and f-measure for these two baseline models, and for several new word-level models introduced below. As shown by the third column, the majority vote word-level model outperforms the accuracy of the turn-level model. Both of these models also significantly ($p < .01$) outperform the 78.70% accuracy of always predicting the most frequent class *nonUncertain*. However, unlike our prior work which only used pitch features [10], now that we have added energy features the difference between the turn and majority word-level models is no longer significant ($p < .6$).

Note that our majority word-level model makes two important assumptions. First, since annotation is only available at the turn-level, all words are given the classification of their parent turns during training. For *Uncertain* turns this assumption contends with our intuitions, e.g. in statements uttered as questions only the last word(s) typically bear prosodic marks of uncertainty. A second assumption which also contends with this intuition is that a simple majority vote can be used to derive the turn class from the predicted word classes. While we have yet to tackle the first assumption, below we show the utility of tackling the second by incorporating a linguistically motivated notion of word quality based on position in turn into the voting method.

4.2. Voting scheme: Oracle

To derive an upper bound for voting scheme improvements, we use an **oracle**: among the candidate votes always pick the correct one. For example, during testing, if the turn is labeled as *Uncertain* and at least one of the words is predicted as *Uncertain*, the **oracle** voting scheme will label the turn *Uncertain* regardless of how many other words in the turn are predicted as *Uncertain* or *nonUncertain*. If none of the words is predicted to be *Uncertain*, the turn is labeled as *nonUncertain*. The fourth column of Table 1 shows that such a "perfect" voting scheme greatly boosts accuracy: a significant 9.09 and 9.16 absolute percentage improvement over the majority voting and turn-level models. Precision and recall figures are also high. These **oracle** results suggest that if we can develop a voting approach that knows which word vote(s) to count, our turn-level classification performance should improve. We thus explore several methods for automatically picking such words, by exploiting linguistic knowledge regarding prosodic cues to uncertainty in English.

4.3. Voting scheme: 1Uncert

Sometimes a turn can sound uncertain if only one word has an uncertain intonation. In fact, our oracle needs to pick only one correctly classified uncertain word, to classify the entire turn as uncertain. We thus hypothesize it may be beneficial to allow a turn to be classified as *Uncertain* if at least one word is predicted to be *Uncertain*. We call this voting scheme **1Uncert**. The fifth column in Table 1 shows that this approach is far too lenient: the accuracy drops to 74.51%, which is below all baselines.

4.4. Voting scheme: lastWord

Based on the failure of **1Uncert**, we hypothesize that to improve our voting mechanism we need to better understand the potential positive contribution of each word to the turn-level classification. One way to examine this is to look at how many times the prediction of different types of words in a turn match the classification of the turn. For example, because uncertain turns are often statements uttered as questions (which are known to be prosodically marked by rising intonation [17, 18]), perhaps voting should use only the last word(s) in a turn. Figure 2 shows the contribution of each word as function of distance from the last word in the turn over all experiments. The graph shows that for 78% of the turns the prediction for the last word matches the turn label; the prediction for the word before the last matches

Table 1: *Performance for turn and word-level models. Significant accuracy improvements ($p < .01$) over baselines marked with a *.*

Model	turn	majority	oracle	1Uncert	lastWord	linWgt	logWgt
Accuracy	83.04	83.11	92.20	74.51	84.26	85.43	83.47
Improvement over turn-level	-	+0.07	+9.16*	-8.53*	+1.22*	+2.39*	+0.43*
Improvement over majority	-0.07	-	+9.09*	-8.60*	+1.15*	+2.32*	+0.36*
Precision	0.629	0.701	0.893	0.440	0.657	0.739	0.686
Recall	0.498	0.361	0.720	0.720	0.547	0.490	0.415
F-Measure	0.556	0.477	0.797	0.546	0.597	0.588	0.516

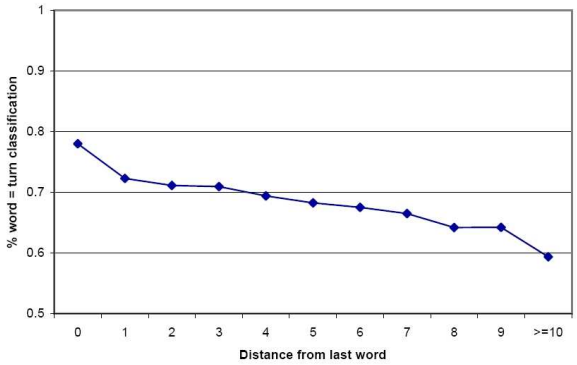

Figure 2: *Word prediction as function of distance.*

the turn label in 72% of the cases. The further we move away from the last word the lower the match percentage, i.e. the model seems to classify words more reliably at the turn's end.

We thus add a constraint to our prior voting scheme, namely that while only one uncertain word is still needed to classify an entire turn as uncertain, the uncertain word must occur at the turn's end. This is our **lastWord** voting scheme. Despite its simplicity, this method works better than majority voting. As Table 1 shows, the last word method yields 84.26% accuracy – a significant 1.15 improvement over majority voting. With a slight decrease in precision (from 0.701 down to 0.657), we also see a large increase in recall (from 0.361 to 0.547).

4.5. Weighted voting schemes: linWgt and logWgt

While the **lastWord** voting scheme shows the importance of the last word in the turn, Figure 2 shows that other final words might also have a positive contribution. This suggests that later words in the turn should have more influence in the voting scheme, but perhaps it would still be of benefit to give the earlier word predictions some attention. Controlled experiments [15] have also suggested that prosodic manifestations of uncertainty occur both locally to particular words or phrases, as well as in the surrounding context. We thus created a weighted majority voting scheme that would give "more votes" to the word-level predictions that appear later in turns. Since it is not clear what weight dynamics fits the descending trend from Figure 2, we experimented with 2 weighted schemes: linear weighting (**linWgt**) and log weighting (**logWgt**).

Suppose we are dealing with a turn with 5 words. Under the old majority voting scheme, each word has 1/5 of the 5 total votes. Under the linear scheme, the first word gets 1 vote, the second word gets 2 votes, and so on. This way, the fifth word will receive 1/3 of the total votes, while the first word only has 1/15. The log weighting scheme works the same way using $log(x)$ votes for the word at position x.

Employing these weighting schemes improves accuracy significantly. While **logWgt** only shows a 0.36 absolute percentage improvement over majority voting, the linear version bests it by 2.32. Due to the slower growth of the log function and the fact that our turns have only 2.8 words on average, the **logWgt** assigns weights closer to the uniform weighting used in majority voting. Thus, it is not surprising that we do not see a big change in performance for **logWgt**. In contrast, the linear weighting boosts performance even more. Its precision is better than majority voting (from 0.701 to 0.739) and its recall is better as well (0.490 compared to 0.361).

4.6. Discussion

Although our majority model improves over the turn-level model as in our prior work, with this paper's addition of energy features the difference is no longer significant. However, we developed several other word-level models that significantly outperform both the turn-level and the majority word-level models. Using the last word as the reference point in our successful voting schemes was not random, but was informed by the dynamics of how uncertainty is prosodically expressed in English. Many examples of uncertainty in our corpus are statements uttered as questions, i.e. with rising intonation, which suggested that later words might have the most significant prosodic change. Our analysis from Figure 2 and the improvements offered by **linWgt** validate this intuition. Consequently, unlike the turn-level model, new versions of our word-level model offer the ability to include prior knowledge regarding the dynamics of uncertainty expression via the voting scheme. While for uncertainty linear voting based on word position was successful, other emotions might require voting based on other linguistic properties.

5. Related work on sub-turn features

Others have also used sub-turn features for emotion prediction. As discussed in Section 1, our choice of words was motivated by our focus on spoken dialogue. [8] similarly examines word-level features in human-computer dialogue data, but does not explicitly compare their informativeness to turn-level features. [19, 9] use voiced-segments, which are similar to words. In contrast, [20] uses a small 50 msec window, while [7] uses bigger breath group units (speech between two pauses).

To handle the many-to-one relationship between sub-turn units and a single turn-level class, [9] uses a weighted voting approach similar to ours; however, weighting is based on sub-turn length rather than position. In contrast, [7] still directly classifies turns rather than sub-turns, but handles the many-to-one problem by only considering sub-turn features from a fixed predefined subset of sub-turns (in particular, the first, last and longest breath groups in a turn, reinforcing the utility of length and position in weighted voting methods). Moving from classi-

fication to sequence and other modeling approaches, [20] combines sub-turn features by fitting parametric distributions then uses the parameters as turn-level features, while [19] treats sub-turn features as observations in a Hidden Markov Model.

In terms of performance comparison between sub-turn and turn-level features, the results are mixed. When parametric models are used, sub-turn features perform worse than turn-level features [20, 19]. In classification-based approaches such as ours and that of [7], sub-turn level features usually outperform turn-level features; [9] is an exception, although their turn and sub-turn features are not equivalent. However, many factors besides sub-turn granularity and combination method also vary. Our work and that of [7] predict uncertainty, while [20, 19, 9] predict 6-8 emotion classes. We predict emotions in human-computer spoken dialogues, while other studies use human-human [7, 19, 9] or artificially elicited data. Corpus size ranges from smaller [20, 9] to considerably larger (this study and [7, 19]). Finally, while most studies look at pitch and energy, other features have also been considered [20, 9].

6. Conclusions and future work

We show that word-level pitch and energy features can outperform turn-level features when classifying uncertainty in spoken dialogues. Building on prior work, we use voting to solve the many-to-one relationship between a variable number of word-level segments and a single turn-level classification. We introduce and evaluate several alternatives to majority voting, motivated by the potential for improvement indicated by an "oracle" method. Our new voting schemes are inspired by the literature on uncertainty as well as empirical analysis of our data. We hypothesize that our use of a linear voting method based on word position works best in our experiments as it reflects the use of rising intonation to express uncertainty in English.

Since our results suggest that emotion expression dynamics play an important role, we would like to evaluate other potentially relevant aspects of words besides turn position (e.g. word length, duration, part-of-speech, frequency) to develop additional linguistically informed voting methods, and to reproduce these experiments for other emotions (e.g. frustration). We would also like to try to learn the voting scheme and/or use different ensemble methods than voting. Removing the assumption that all words are labeled using the turn's label during training is another potential avenue for improvement.

7. Acknowledgements

This research is supported by NSF Grant No. 0328431. We thank the ITSPOKE group for feedback, and J. Liscombe and J. Hirschberg at Columbia University for annotations.

8. References

[1] R. Cowie and R. R. Cornelius, "Describing the emotional states that are expressed in speech," *Speech Communication*, vol. 40, no. 1-2, pp. 5–32, 2003.

[2] J. Ang, R. Dhillon, A. Krupski, E.Shriberg, and A. Stolcke, "Prosody-based automatic detection of annoyance and frustration in human-computer dialog," in *Proc. Intl. Conference on Spoken Language Processing*, 2002.

[3] C. M. Lee, S. Narayanan, and R. Pieraccini, "Combining acoustic and language information for emotion recognition," in *Proc. Intl. Conference on Spoken Language Processing*, 2002, pp. 873–876.

[4] J. Liscombe, G. Riccardi, and D. Hakkani-Tur, "Using context to improve emotion detection in spoken dialog systems," in *Proc. Interspeech*, 2005.

[5] B. Schuller, S. Steidl, and A. Batliner, "The Interspeech 2009 emotion challenge," in *Proc. Interspeech*, 2009.

[6] M. Rotaru and D. J. Litman, "Using word-level pitch features to better predict student emotions during spoken tutoring dialogues," in *Proc. Interspeech*, 2005.

[7] J. Liscombe, J. Venditti, and J. Hirschberg, "Detecting certainness in spoken tutorial dialogues," in *Proc. Interspeech*, 2005, pp. 1837–1840.

[8] A. Batliner, K. Fischer, R. Huber, J. Spilker, and E. Noth, "How to find trouble in communication," *Speech Communication*, vol. 40, no. 1-2, pp. 117–143, 2003.

[9] M. Shami and W. Verhelst, "An evaluation of the robustness of existing supervised machine learning approaches to the classification of emotions in speech," *Speech Communication*, vol. 49, no. 3, p. 201, 2007.

[10] G. Nicholas, M. Rotaru, and D. J. Litman, "Exploiting word-level features for emotion prediction," in *Proc. IEEE/ACL Workshop on Spoken Language Technology*, 2006.

[11] D. Litman and S. Silliman, "ITSPOKE: An intelligent tutoring spoken dialogue system," in *Proc. Human Language Technology/North American Chapter of the Association for Computational Linguistics (Companion Vol.)*, 2004, pp. 233–236.

[12] K. VanLehn, P. W. Jordan, C. Rosé, D. Bhembe, M. Böttner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, S. Siler, R. Srivastava, and R. Wilson, "The architecture of Why2-Atlas: A coach for qualitative physics essay writing," in *Proc. Intl. Intelligent Tutoring Systems Conference*, 2002, pp. 158–167.

[13] H. Pon-Barry, K. Schultz, E. O. Bratt, B. Clark, and S. Peters, "Responding to student uncertainty in spoken tutorial dialogue systems," *Intl. Journal of Artificial Intelligence in Education*, vol. 16, pp. 171–194, 2006.

[14] K. Scherer, "Vocal communication of emotion: A review of research paradigms," *Speech Communication*, vol. 40, no. 1-2, 2003.

[15] H. Pon-Barry, "Prosodic manifestations of confidence and uncertainty in spoken language," in *Proc. Interspeech*, 2008.

[16] P. Taylor, "Analysis and synthesis of intonation using the tilt model," *Journal of the Acoustical Society of America*, vol. 107, 2000.

[17] J. Liscombe, J. J. Venditti, and J. Hirschberg, "Detecting question-bearing turns in spoken tutorial dialogues," in *Proc. Interspeech*, 2006.

[18] C. Dijkstra, E. Krahmer, and M. Swerts, "Manipulating uncertainty: The contribution of different audiovisual prosodic cues to the perception of confidence," in *Proc. Speech Prosody*, 2006.

[19] B. Schuller, G. Rigoll, and M. Lang, "Hidden markov model-based speech emotion recognition," in *Multimedia and Expo (ICME)*, 2003, pp. 401–404.

[20] Y. Li and Y. Zhao, "Recognizing emotions in speech using short-term and long-term features," in *Proc. Intl. Conference on Spoken Language Processing*, 1998.