Ethan Hartzell

Toward a Feature-Based POS Tagger for ASL

Abstract

American Sign Language (ASL) presents unique challenges to NLP in both the domains of recognition and POS tagging due to the simultaneous nature of its linguistic features. Using the NCSLGR corpus, which contains 1887 utterances annotated for not only glosses and POS tags but also a wide array of nonmanual features, we tokenize signs as pairs of glosses along with pairs of POS tags (one for each hand). Splitting these utterances into training and test sets, we train a bigram HMM model and a unigram Maximum Entropy Markov model using both manual and nonmanual linguistic features, and compare them to a Most Common Class baseline. The bigram HMM performs consistently worse than both the unigram MEMM and the baseline in all cases. The MEMM outperforms the baseline only when using manual features or both manual and nonmanual features. Future work should aim to explore better ways to encode nonmanual features so that they may be more useful in predicting a tag.

Introduction

American Sign Language (ASL) is an understudied language in NLP, despite strong interest from linguists in its implications for linguistic theory. POS tagging is a problem that has been more or less solved for standard English using Hidden Markov Models; however, due to the lack of a standard written form, ASL has not been subjected to rigorous study. Additionally, ASL presents other unique challenges, such as that phonemes are realized simultaneously rather than sequentially. In fact, signs can sometimes be produced simultaneously as well, such that the dominant and non-dominant hands may simultaneously have different POS tags.. The number of possible phonemes in ASL is approximately 1.5×10^9, which makes ASL challenging for computer vision recognition (Vogler & Metaxas 2001). An accurate POS tagger for ASL could help performance on other tasks, such as machine translation, and be of use to linguists in future research.

Dataset

For the POS-tagging task, I used the National Center for Sign Language and Gesture Resources (NCSLGR) Corpus which contains a total of 1887 utterances and 12571 tokens. The corpus is broken up into short narratives of several different genres by four different signers and demonstrates a wide variety of grammatical structures. This corpus presents its own challenges as well. It was extensively annotated by linguists – video files are split up to include glosses for dominant and non-dominant hands along with POS tags for each hand, and many non-manual features of the sign too. The way I tokenized signs was to treat both glosses together as a single unit, tagged with both POS-tags also as a single unit. Because the tagset used by the annotators included 29 tags, this returns a possible 841 tags, though in reality not every tag combination is possible. The average tagset observed in a training set is between 100 to 200 tags.

The utterances used were split into training and test sets by first randomizing the sentence order, to account for vocabulary and genre differences between the various narratives in the corpus, and in one experiment the training set used 300 of these sentences where the test set used 100; in another, both had 300 sentences. To give an idea of how a token in represented in this model, see the sentence below: “How many books has the student read so far?”

token: ((u'LEARN+AGENT', 'null'), (u'Noun', 'null'))
token: ((u'UP-TO-NOW', 'null'), (u'Tense+Aspect', 'null'))
token: ((u'FINISH', 'null'), (u'Verb', 'null'))
token: ((u'READ_2', 'null'), (u'Verb', 'null'))
token: ((u'HOW-MANY/MANY', 'null'), (u'Quantifier+Wh-word', 'null'))
token: ((u'BOOK+', 'null'), (u'Noun', 'null'))
token: ((u'part:indef', 'null'), (u'Particle', 'null'))

I have replaced instances where the non-dominant hand is not a part of the sign with the string “null”. The full corpus can also be accessed online.

Experimental method

Preprocessing:

First the XML files containing the corpus are parsed using Christian Vogler’s Signstream-XML parser, available online. Then, the annotations are tokenized on the basis that the dominant and nondominant hand glosses represent the token, and the pair of tags for each hand represent the tag for the token. Because the corpus contains narratives from many different genres (e.g. a scary story and a motorcycle accident), the order of sentences is randomized to avoid a large number of unknown tokens. A static vocabulary is selected from the tokens that occur at least twice in the training set. Start and end tokens are added to each sentence, and unknown tokens replace those that are not in the vocabulary. Next, tokens are converted into feature sets for MaxEnt. Also note that gesture tokens were treated as instances of the same token.

Feature Selection:

In all trials, features representing a token’s identity, whether it is a certain word in the vocabulary or not, is embedded.

In some trials, features representing manual phonological and morphological aspects of the signs are used. These features are mainly based on the SignStream glossing conventions. The manual features are as follows: whether a sign was finger-spelled, whether a sign was reduplicated one, whether a sign was reduplicated twice, whether a sign contained overt spatial agreement, whether the gloss contained “IX,” “POSS,” or “SELF,” whether the gloss contained “loc,” “dir,” or “arc,” whether the sign used the continuative aspect, and whether the sign used the reciprocal aspect.

In some trials, features representing nonmanual phonological and morphological aspects of the signs are used. These features are based on whether or not the nonmanual features overlap with the main gloss in its start and end times. The nonmanual features are as follows: negation via headshaking, WH-question via furrowed brows, yes-no question via raised brows, conditional/when clauses via a number of nonmanual features, rhetorical questions via facial expression, topicalization via facial expression and movement, and role shift via body movement.

In some trials both nonmanual and manual features were used together. I have tried to select features that are only plainly observable and that do not depend on linguistic analysis – for example the nonmanual feature of "topic/focus" lists several fields depending on the sign's position in the sentence, but I opted to only use whether or not there was an overlapping topic/focus annotation for any field as this feature. The SignStream annotators note that conditional vs. when clauses may actually be indicated by the same facial expression, but that it is up for debate (Neidle 2002). I have used them as separate features, but if they are indeed arising from the same combination of nonmanual features then it would be much more difficult to extract them as separate features from a video file and perhaps less useful in future attempts at POS tagging.

Main Procedure:

In all trials, we instantiate and Maximum Entropy Classifier using NLTK tools as well as train our own bigram hidden Markov model. We run four trials for each experiment, one without manual or nonmanual features, one with manual features, one with nonmanual features, and one with both. We report tag accuracies, percent ambiguity of the training set, size of the tagset observed in the training set, and the size of the vocabulary.

How to Run:

To run the experiment with both manual and nonmanual features, simply run "python MEMM_tagger.py" and wait for the results. This script relies on NLTK and Vogler's SignStream XML Parser, which can be downloaded for free along with the corpus at this address: http://www.bu.edu/asllrp/ncslgr-for-download/download-info.html

If the XML parser and the corpus do not fit in my provide submission, I recommend putting the MEMM_tagger.py file inside the SignStream XML parser file. Then, you may need to adjust the path on the line 490 to be the location of the XML files containing the corpus. Note that this implementation is rather time expensive, so do not try to increase the size of the training corpus more than is already specified. As is, the program will take about five minutes to train and test.

Evaluation

For evaluation, we compare the accuracies of our unigram MEMM with those of a bigram HMM and a Most Common Class baseline. See below the detailed results of both experiments. Tag accuracies are percentages.

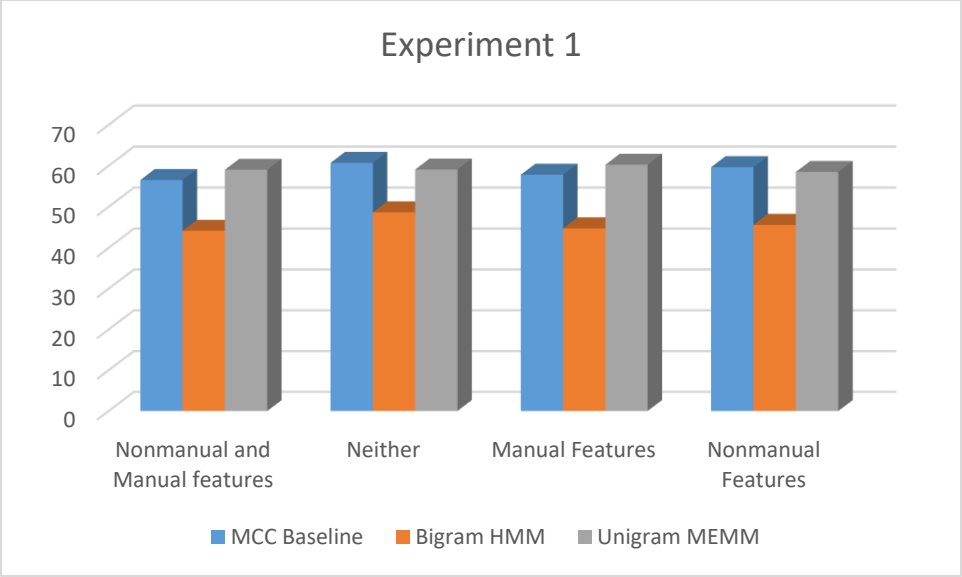
Experiment 1:

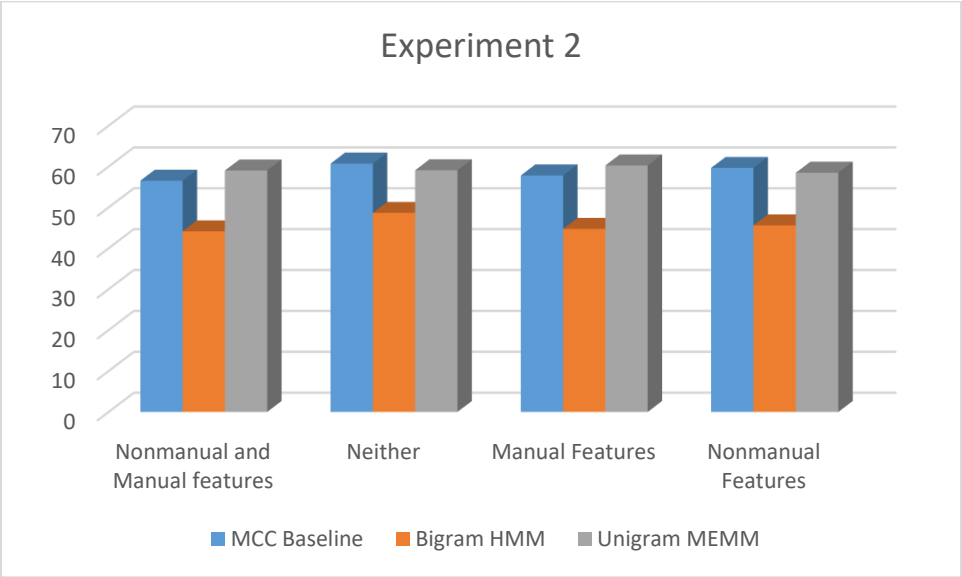
Trial	Training/Test Split	Vocab Size	Tagset Size	Percent Ambiguous	MCC Baseline	Bigram HMM	Unigram MEMM
Manual and nonmanual features	300/100 sent	274	43	45.61	56.98	45.96	**58.54**
No features	300/100 sent	268	47	46.15	**56.50**	45.85	49.40
Manual features	300/100 sent	275	48	44.61	59.90	49.77	**62.78**
Nonmanual Features	300/100 sent	289	63	46.76	**61.80**	49.84	54.81

Experiment 2:

Trial	Training/Test Split	Vocab Size	Tagset Size	Percent Ambiguous	MCC Baseline	Bigram HMM	Unigram MEMM
Manual and nonmanual features	300/300 sent	251	55	47.21	56.54	44.15	**59.04**
No features	300/300 sent	281	49	48.80	**60.73**	48.66	59.10
Manual features	300/300 sent	250	51	46.43	57.83	44.74	**60.27**
Nonmanual Features	300/300 sent	257	54	46.17	**59.66**	45.60	58.51

To better visualize these results, see the graphs below.





The results of the second experiment closely mirror that of the first. In all cases, the unigram MEMM and the baseline outperform our bigram HMM. This may mean that tag transitions are not good predictors of a tag for this dataset, or that long-range dependencies are more important to ASL syntax. Whatever the reason, a bigram HMM misses a great deal of information about the sign, which is crucial for outperforming the baseline. Our unigram MEMM outperforms the baseline in two conditions, with nonmanual and manual features, and with just manual features. This could suggest that manual features are stronger predictors of a token's part of speech than are nonmanual features. However, it could also be that the nonmanual features were poorly selected. A sign was considered to have a nonmanual features only if the annotation time for the nonmanual feature overlapped with that of the dominant hand gloss; that is, the actual start and end times did not come into consideration, which could be

significant because it has been shown that prosodic units correspond with syntactic constituents (Sandler 2010).

From analyzing the confusion tables we find some surprising results with regards to the reported accuracies. The table below shows the number of times each model mistook nouns and verbs. It is expected that the MEMM would perform better on this specifically because uses features like directional agreement and reduplication that are specific to verbs, but it is surprising that the bigram HMM vastly outperforms in this regard since it in no case outperforms the baseline.

MCC	Bigram HMM	Unigram MEMM
254	10	157

Conversely, with regards to confusing determiners and pronouns, the models perform in a way that reflects the reported accuracies. The MEMM outperforms both, but only barely outperforms the baseline. The word which causes the most confusion for these two classes is "IX-3P:i" which sometimes means he/she/it, but other times functions as a determiner. For some reason, a bigram model is actually worse at tagging these.

MCC	Bigram HMM	Unigram MEMM
24	38	23

Conclusions

Future research into the domain should seek to better understand the reasons for the bigram HMM's poor performance, perhaps by comparing to n-gram models of higher orders. Likewise, whether or not higher order n-gram MEMM's could improve performance should also be explored. More in-depth research into feature selection, especially regarding how to improve the use of non-manual features, is likely to benefit future taggers. Vogler & Metaxas (2001) demonstrated that paHMMs (parallel Hidden Markov Models) modeling the simultaneous aspects of ASL phonology independently returned higher accuracies for computer vision recognition of ASL than traditional HMMs. It is possible that extending paHMMs for part of speech tagging may also return better results, although it is clear from our data that feature-based taggers traditional bigram HMMs. Therefore, it is predicted that the optimal tagger would be one that combines the two to create paMEMMs. Another possible improvement lies in that MEMMs do not implicitly model interactions between features. The use of classifiers that do may also be of use in creating a feature-based tagger. Another limitation of MEMMs is that they are prone to overfitting. It is possible that our tagger's accuracy could have been improved by some type of regularization.

References

Vogler, C. and Metaxas, D., 2001. A framework for recognizing the simultaneous aspects of american sign language. *Computer Vision and Image Understanding, 81*(3), pp.358-384.

Neidle, C . SignStream™ Annotation: Conventions used for the American Sign Language Linguistic Research Project. . (August 2002). http://www.bu.edu/asllrp/asllrpr11.pdf

Sandler, W., 2010. Prosody and syntax in sign languages. Transactions of the Philological Society, 108(3), pp.298-328.