information between adjacent tokens in the data. This corresponds to maximizing the likelihood of the data assuming a bigram model.

The benefit of using word classes is that the resulting language model can have fewer parameters than a word-based n-gram model. In our case, the in-domain training text is so small that it can be a real advantage to have fewer parameters to learn in the language model. The problem is that if the data is too small to reliably estimate word transition probabilities ($p(w_i|w_{i-1})$), then it will also be difficult to learn a good partitioning of the words into classes.

Our hypothesis is that the advantage in learning the word class assignments on the Twitter data, which solves the data sparsity problem, outweighs any performance penalty that is incurred due to domain mis-match. This differs from the traditional approach where the class assignments ($w_i \in c_j$) are learned from the same training text which is used to estimate the class transition probabilities ($p(c_i|c_{i-1})$) and the word probabilities ($p(w_i|c_j)$). Twitter data, in these experiments, refers to the concatenation of the text downloaded from Twitter with our in-domain data for the reasons described above. The experiments in Section IV-C will compare learning the class assignments on out-of-domain data (hybrid method) to the traditional approach (baseline).

## IV. Experiments

### A. Experimental Data

The in-domain data used in the experiments in this paper comes from the IARPA Babel program.[1] This program focuses on keyword search for low-resource languages. The languages are low resource in the sense that they have fewer native speakers than the languages receiving the most attention from researchers and also in the sense that the provided training data is small in comparison to what is typically used. We exclusively focused on the so-called limited language pack, which consists of only ten hours of recorded telephone conversations. Our experiments were conducted using the Bengali, Tamil, Turkish and Zulu languages. The languages that we selected for our experiments have the largest vocabulary sizes (See Table I) of the languages in the Babel program and thus suffer the most from the data sparsity problem. As a point of comparison, Tagalog, which was not used in our experiments, has one third as many vocabulary items as Tamil.

[1] http://www.iarpa.gov/index.php/research-programs/babel

TABLE I
Language Vocabulary Size

| Language | Types | Tokens |
|---|---|---|
| Bengali | 7,932 | 72,614 |
| Tamil | 14,264 | 70,258 |
| Turkish | 10,069 | 67,362 |
| Zulu | 13,628 | 58,027 |

### B. N-gram Language Models

We performed data collection and language model training experiments on the Bengali, Tamil, Turkish and Zulu languages. We were able to collect useful data for each of the four languages as seen in Table II. The data collection experiment was especially successful for Turkish and Bengali. For both of those languages the interpolation weight given to the Twitter LM was over 20% and the corresponding reduction in perplexity was more than 12%.

TABLE II
Twitter Data Collection Experiment Results

| Lang. | Users | Lines | $\Delta$ PPL | Weight |
|---|---|---|---|---|
| Bengali | 12k | 7.7M | 12.5% | .21 |
| Tamil | 13k | 4.7M | 7.5% | .18 |
| Turkish | 7k | 13.6M | 13.6% | .27 |
| Zulu | 3k | 5.2M | 5.1% | .11 |

A large amount of data was also collected for Zulu and Tamil although the perplexity reduction was not as large as it was for Bengali and Turkish. There are a few possible explanations for this outcome. Both Zulu and Tamil have larger vocabularies (See Table I) than the other languages, which