ABSTRACT
A natural dialogue system for human-computer interactions includes an understanding module that defines groups of words and phrases that are semantically similar. New domains usually do not have large, annotated corpora, so it is useful to develop methods of automatically inducing semantic groups (concepts). Classes can be induced from unannotated corpora by means of a context-dependent similarity measure, such as the Kullback-Leibler distance. However, the precision of auto-induced classes is reduced in cases where the statistics are poor, or where words of different parts of speech may occur in similar lexical contexts. We address this issue by augmenting a semantic generalizer with three new modules, a part-of-speech (POS) tagger to preprocess the list of candidate word pairs, trigram instead of bigram contexts, and context thresholding. The subjective quality of auto-induced classes is compared for these three methodologies for a large newspaper text (WSJ) corpus. We show that context thresholding has the biggest impact on inducing higher quality classes. The best results were obtained for a context threshold of 3 extant bigrams and trigrams. For bigram contexts, with POS tags, the precision was 88% for the first 50 clusters, 75% for the first 100 clusters, and 65% for the first 150. Similar results were attained for trigram contexts and no POS tags.

1. INTRODUCTION
A developer can generate groups of semantically related words manually, but this is a time-consuming process [1, 2]. Some classes, such as those consisting of lists of names, are easy to specify, whereas others require a deeper understanding of language structure. We and others have shown that statistical processing techniques can be used to semi-automatically generate concepts from unannotated corpora [3, 4, 5] for a single domain because semantically similar phrases often share similar syntactic environments for limited domains [6, 7]. An iterative procedure is typically used to successively generate groups of words and phrases with similar semantic meaning from a corpus consisting of training sentences. This procedure has been tried on human-machine dialogues for small tasks such as travel information, but not on a large corpus such as the Wall Street Journal.

Semantic classes can be auto-induced using a similarity metric [3]-[5]. The choice of the metric used to determine the degree of similarity between two candidate words being considered for a semantic class is a critical issue. In our earlier work [5], we compared the quality of induced classes using four different distance metrics, one of which was the Kullback-Leibler (KL) metric. In that study, we examined classes generated for three small corpora, as well as a larger text-based corpus taken from Wall Street Journal (WSJ) articles. The poor quality of the classes induced from the WSJ corpus led us to consider three new methodologies to enhance our semantic classifier: including part-of-speech (POS) tags, introducing trigram contexts, and context thresholding (a constraint on the n-gram statistics). We compare the precision of auto-induced classes for different combinations of these three methods when used in conjunction with the KL metric. The metrics are evaluated using about 7000 sentences taken from the Wall Street Journal, a text-based corpus. The metrics are evaluated by comparing results from automatic and manual annotation of semantic classes.

This work is part of a larger project to automatically build semantic classes that can be used to port a concept from one domain to another. We have previously shown that two metrics, the concept-comparison and concept-projection metric, can be used to measure the degree of domain independence of a particular concept class [2]. The ability to automatically induce a concept in one domain and port it to a new domain for which little training data is available would be a powerful tool for developers building new speech services.

2. AUTO-INDUCTION OF CONCEPTS
There are two major issues when auto-inducing classes: 1) finding phrases that act as a single lexical unit, and 2) finding words (and phrases) with similar semantic content and then grouping them into semantic classes or concepts. The second issue is the focus of this study.

Concepts are auto-induced in an iterative process, shown schematically in Fig. 1. On the left it is shown how a simple sentence, in this example taken from a travel corpus, is processed by each module. There are three main steps to auto-inducing classes, a lexical phraser which groups words in a single lexical unit, a semantic generalizer that generates rules that map words (and concepts) to concepts, and a corpus parser which re-parses the corpus using the rules generated from the semantic generalizer.

2.1. Lexical phraser
The top block in Fig. 1 is the lexical phraser that creates a list containing common phrases, or sentence-fragments. Frequently co-occurring words such as “New York” are chunked into a single phrase, e.g., New York → [NewYork]. Furthermore, we induced hierarchical phrasing by permitting the phraser to operate on its own output.

The lexical phraser groups consecutive words into phrases by using a weighted point-wise mutual information (MI) measure [8] to find those lexical entities (referred to as words in the remainder of this paper) that are semantically related.
of this paper) that co-occur often. The \( n \) phrases with the largest MI measure,

\[
MI(w_1, w_2) = p(w_1, w_2) \log \frac{p(w_1, w_2)}{p(w_1)p(w_2)}
\]

(1)

for the words \( w_1 \) and \( w_2 \), are kept at each iteration. They are only retained in successive iterations if they are classified into semantic groups in the following, semantic generalizer, module.

We found in our earlier studies that \( n=30 \) phrases (or more generally, \( \text{chunks} \)) per iteration were a reasonable number. Fewer than 10 chunks meant that certain commonly occurring phrases, such as \( \text{I want} \), would not be combined, but more than 50 chunks created so many nested chunks, such as \([\text{go to New York}]\), that entire sentences were frequently combined into a single sentence entity in the smaller domains. This prevented further semantic generalizations for words or sentence fragments (such as \( \text{Newark} \) being a member of a \( \langle \text{city name} \rangle \) class, where brackets denote a semantic class label) within these large sentence-level chunks.

2.2. Semantic generalization

The next block in Fig. 1 is the semantic generalizer. Grammar rules are generated each iteration, where a rule maps a word, sentence fragment (from the previous block), or previously formed class, into a semantic class whose members share the same meaning. The main criterion for selecting such groupings is the similarity of the left or right-hand context for the members of a group. For example, city names are grouped into the same class because they are used in similar lexical contexts, for example: \( \langle \text{city name} \rangle \) tomorrow. Only one semantic merger should be generated each iteration so that the new semantic group can be incorporated into the corpus immediately. To speed up the computation, five rules were generated per iteration; no qualitative difference was seen.

In the next sections we discuss the different conditions used in our studies. We compare the subjective quality of induced classes for the case where the corpus is tagged using a part-of-speech (POS) tagger and for the cases where bigram and trigram contexts are used in the determination of lexical similarity.

2.3. Brill Tagging

In this study, we compare the subjective quality of classes induced by several different methods. One of these methods includes using a POS tagger to tag each word in a corpus with the appropriate part of speech. POS tagging is a process that assigns a lexical class marker to each word of the corpus. Since the inception of the speech understanding and analysis, people have manually created rules for tagging. There are two steps, the first steps is to mark a word based on a pre-constructed dictionary. A set of disambiguation rules are used to select the candidate tags. These rules are derived from the corpus. In order to automate the tagging process, Jelinek [9] proposed a Markov model based stochastic tagger. This tagger assigns a tag \( t_1 \) to word \( w_1 \) based on the maximization of the conditional probability product, \( p(w_1 | t_1)p(t_1 | t_2, t_3, ...) \). Although the stochastic methodology alleviates the burden of manually constructing rules, it suffers from an incomplete capture of linguistic information. In 1992, Brill proposed a trainable rule-based tagger [10]; not only does it achieve comparable performance as the stochastic tagger, but it also encodes the linguistic information directly into the rules. Each word in a corpus is assigned a tag from the set of Penn Treebank POS tags, \( \text{e.g.}, \text{VBD for verbs in past participle form and CC for conjunctions.} \)

We used the POS tagger to annotate each word in a corpus consisting of about 7000 sentences taken from the WSJ news articles. These tags are used in two ways. First, the language model uses the statistics of the tagged words when calculating the \( n \)-gram probabilities. Therefore, a word that occurs with two different POS senses is treated as two separate words. Second, the tags are used in a pre-filtering process. Only those word pairs which have the same tags are added to the pair list of words for consideration as candidates to be merged into the same class.

2.4. Generalizing in bigram and trigram contexts

The semantic generalizer pairs words or phrases (generated in the preceding lexical phaser module) according to the similarity of their syntactic environments. We consider a candidate word, \( w \), in a word sequence,

\[
v_1^L v_1^R w v_2^L v_2^R
\]

(2)

with \( v_1^L \) and \( v_1^R \) words in the left context and with \( v_2^L \) and \( v_2^R \) words in the right context. For bigram contexts, two probability distributions are calculated, \( p^L (v_1^L | w) \) and \( p^R (v_2^R | w) \), for the the left and right bigram contexts respectively. The right-context bigrams are calculated using the usual word order, and the left-context probabilities are calculated with a reversed order training corpus using standard \( n \)-gram training tools.

Our earlier studies showed that the bigram-context may not be sufficient to auto-induce classes in open-ended domains such as the WSJ corpus. In these studies bigram contexts with no POS tags or context thresholding were used. About 80% of the first
100 auto-induced class members were evaluated by human subjects to be wrong! A typical example was the cluster \{airport, seventeenth\}. The most common lexical context that contained these words was \{...the \( <\text{airport} > \), where \( <\text{airport} > \) is the end of sentence marker. This indicated the bigram context was sometimes too local to capture semantic similarity. This led us to consider using trigrams for the lexical context when calculating the distances between pairs of words. Extending the calculations using bigram contexts, we calculate the two probability distributions, \( p^R(v_i^R v_i^R | w) \) and \( p^R(v_i^R v_i^R | w) \) for the trigram contexts.

We estimate the similarity of two words, \( w_1 \) and \( w_2 \), as the sum of the symmetric left and right context-dependent distances \cite{7}, giving the total distance,

\[
D^{LR}(w_1, w_2) = D_{L2}^R + D_{R2}^R + D_{L2}^L + D_{R2}^L
\]

where \( D_{L2}^R = D(p_i^R \parallel p_i^R) \) is the left-context distance and the \( D_{R2}^R \) distance terms are similar, using right-context probabilities.

The Kullback-Leibler (KL) distance has frequently been used for this distance \( D \) when auto-inducing semantic classes. The KL distance, a relative entropy measure \cite{11} of the distance between two distributions, \( p_i \) and \( p_2 \), is unbounded since it includes ratios whose denominators may approach zero. This has the consequence that the KL distance can be dominated by a few, or even just one, terms. This is especially an issue for our studies since we are interested in developing language models for new domains for which there are limited training data and a metric should not be used if the final sum is dominated by one or two terms. This inspired us in an earlier work \cite{5} to compare the quality of induced classes using three other metrics \cite{6}. Although the other metrics performed slightly better, in this study we used the more common KL metric. In the next section, we expand our original definition of the KL metric in terms of bigram probability distributions to include the trigram probabilities of words.

2.4.1. KL distance for the bigram and trigram contexts

Following Eq. 3, the total, symmetric KL distance is given by \( K^{LR}(w_1, w_2) = K_{L2}^R + K_{R2}^R + K_{L2}^L + K_{R2}^L \). As a representative example, the right-context dependent KL distance \( K_{L2}^R \) between two candidate words, \( w_1 \) and \( w_2 \), is defined over the vocabulary \( V \), for the trigram case,

\[
K_{L2}^R = K(p^R(v_i^R v_i^R | w_1) \| p^R(v_i^R v_i^R | w_2)) =
\sum_{v_1^R \in V} \sum_{v_2^R \in V} p^R(v_1^R v_2^R | w_1) \log \frac{p^R(v_1^R v_2^R | w_1)}{p^R(v_1^R v_2^R | w_2)}
\]

where the sums are over all words in the vocabulary, \( V \). Noting the identity, \( p(v_i^R v_i^R | w) = p(v_i^R | w) p(v_i^R | w) \), gives the distance,

\[
K_{L2}^R = \sum_{v_1^R \in V} p^R(v_1^R | w_1) \log \frac{p^R(v_1^R | w_1)}{p^R(v_i^R | w_2)} + \sum_{v_2^R \in V} p^R(v_2^R | w_2) \log \frac{p^R(v_1^R | w_2)}{p^R(v_2^R | w_2)}
\]

where the first term is the familiar bigram context, the only term used when considering the bigram lexical context. The second, trigram, term is used in addition to the first term when considering trigram contexts.

2.4.2. Context thresholding

Our initial studies showed that the trigram context had almost no impact on the precision of auto-induced classes. This was because the statistics were insufficient for both corpora studied. Therefore, we compared the precisions obtained when using context thresholding: words were only added to the word-pair candidate list when a minimum number of bigram and trigram contexts occurred. As an example, a threshold of “three” means that a word must have at least three extant bigrams and trigrams. This eliminates singletons, such as cases where the human subject uses a single word to answer a system query, and in general, restricts candidates to those words well-represented in a broader lexical context.

3. EXPERIMENTAL RESULTS

3.1. Experimental design

The WSJ corpus is fairly open-ended, covering many topics; articles range in size from about 50 words to several thousand words. We used a corpus consisting of 152,526 words in 6,920 sentences. There were 13,219 unique words (unigrams), 11,441 bigrams and 6,484 trigrams. Bigrams and trigrams were only included for extant word sequences. A cutoff threshold of three was used when calculating the bigram and trigram probabilities.

POS tags were also added to the corpus; in adding the tags, the language model is altered because a word with different POS senses will be considered as separate words — the statistics for each word/POS pair will be calculated separately. The word “executive,” for example may be considered as two separate words (as in executive/NN vice/NN president/NN, or senior/JJ executive/NN). Only those words with the same POS tag are considered as candidates for a merger.

We also applied a unigram threshold in all the studies here, but the minimum required extant bigrams and trigrams (context thresholding), for both right and left contexts, were varied from 0 to 3. Thus, the minimum number of extant n-grams was also used as a separate pre-filter for the candidate pair list.

In a subjective comparison, three human subjects evaluated the quality of the first 150 terminal rules\footnote{Terminal rules are rules that do not group classes into other classes (seventy \( \Rightarrow <G0> \) is a terminal rule, but \( <G1> \Rightarrow <G0> \) is not).} for the WSJ domain. Each rule was marked with a 0 (bad rule) or 1 (good rule). Ratings were aggregated as more rules were added; the scores from the three subjects were averaged every 10 rules.

3.2. Evaluation of derived classes

Table 1 shows the first ten classes induced in the WSJ domain using the KL similarity metric for bigram contexts. No POS tags were used and a context threshold of 3 required candidate words (phrases) to have at least 3 extant bigrams and trigrams. The classes shown are after 100 terminal rules at which point a total of 34 different semantic classes had been generated.

Fig. 2 plots the precision, over a range of 10% to 100%, of the auto-induced semantic rules against the number of rules generated. Three sets of data are shown. The BX labels indicate experiments with bigram contexts and no POS tags, BP refers to bigram contexts with POS tags, and TX refers to trigram contexts without POS tags. At the time of this study, no data had been collected using trigram contexts and POS tags. The pairs of numbers shown in
the inset labels are the bigram and trigram context thresholds; for example, “11” refers to data taken when at least 1 extant bigram and 1 extant trigram were required in order for a word pair to be considered as merger candidates.

The precision generally decreases with increasing number of rules as the closest pairs of words and phrases are removed from the corpus. This means that the raters found that rules produced in initial iterations of the semantic clustering were of generally higher quality than later rules.

The largest effect on the quality of the rules was introducing the context thresholding; a thresholding of three extant bigrams and trigrams (square lines in Fig 2) outperformed the lower thresholds of one and zero. This suggests that it is critical to gather enough statistics to determine the similarity between words; using completely backed-off probabilities in the KL distance measurement leads to fallacious groupings.

The precision of the POS tagged data (solid lines) was about 20% higher than that for the non-tagged data for the case where no extant bigrams or trigrams were required. However, in the cases where extant thresholds for bigram and trigram contexts were required, the precision of induced classes was independent of the POS tagging. This indicates that if extant bigrams and trigrams are required, in both the right and left contexts, then the lexical context is sufficient, and implicitly acts as a type of POS tagger.

Finally, the trigram statistics (dotted lines) only improved performance marginally (5–10%) over the corresponding bigrams (dashed lines) in the cases where one or more extants were required; it is likely that with our small subset of the WSJ corpus the trigram statistics were grossly undertrained. Future work with larger corpora will explore this statistic more carefully.

### 4. CONCLUSIONS

We conclude that context thresholding has the greatest impact on the precision of auto-induced classes from an open-ended corpus such as news articles in the WSJ. The precision of the first 100 terminal rules increases from 20% for no thresholding to 75% when at least three extant bigrams and three extant trigrams are required. POS tags appear to improve precision, especially for lower levels of context thresholding. This would be of greatest value when the training corpus is small. Perhaps surprisingly, trigram contexts had almost no improvement over the bigram contexts, although the small corpus size probably contributed to this result. Future work will investigate the trigram versus bigram precisions using larger corpora since this may be dependent on having better n-gram statistics.

### 5. REFERENCES


