A COMPARISON OF FOUR METRICS FOR AUTO-INDUCING SEMANTIC CLASSES

Andrew Pargellis, Eric Fosler-Lussier, Alexandros Potamianos, Chin-Hui Lee†

Dialogue Systems Research Dept., Bell Labs, Lucent Technologies Murray Hill, NJ, USA
{anp,fosler,potam}@research.bell-labs.com

ABSTRACT

A speech understanding system typically includes a natural language understanding module that defines concepts, i.e., groups of semantically related words. It is a challenge to build a set of concepts for a new domain for which prior knowledge and training data are limited. In our work, concepts are induced automatically from unannotated training data by grouping semantically similar words and phrases together into concept classes. Four context-dependent similarity metrics are proposed and their performance for auto-inducing concepts is evaluated. Two of these metrics are based on the Kullback-Leibler (KL) distance measure, a third is the Manhattan norm, and the fourth is the vector product (VP) similarity measure. The KL and VP metrics consistently underperform the other metrics on the four tasks investigated: movie information, a children's game, travel reservations, and Wall Street Journal news articles. Correct concept classification rates are up to 90% for the movie task.

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. Recently, several studies have shown that statistical processing techniques can be used to semi-automatically generate concepts from unannotated corpora [3, 4, 5, 6] for a single domain because semantically similar phrases often share similar syntactic environments for limited domains [7, 8, 9]. 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 task such as travel information, but not on a large corpus such as the Wall Street Journal.

This work was motivated by a desire to automatically build semantic classes, or concepts, that can be used directly by a natural language understanding system. We have previously shown that domain-independent concepts often occur in similar syntactic and lexical contexts *across* domains [2]. Two metrics, the *concept-comparison* and *concept-projection* metric, were used to measure the portability of a concept from one domain to another. 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.

†Chin-Hui Lee is currently a visiting professor at the School of Computing, National University of Singapore, Singapore (e-mail: chl@comp.nus.edu.sg).

In [3]-[9], the idea of auto-induction of semantic classes using a similarity metric was proposed. The choice of the metric used to determine the degree of similarity between two candidate words being considered for a semantic class is clearly a critical issue. In this paper, we compare the performance of four different metrics used for auto-induction. These metrics are the *Kullback-Leibler* distance, the *Information-Radius* distance, the *Manhattan-Norm* distance, and the *Vector-Product* similarity [6, 10]. The metrics are evaluated for four different application domains: a movie information retrieval service, the Carmen-Sandiego computer game, a travel reservation system, and the Wall Street Journal. The WSJ was a large, text-based corpus. The other three were small, transcribed dialogues between human subjects and agents. The metrics are evaluated by comparing results from automatic and manual annotation of semantic classes.

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, referred to as semantic classes or concepts. The second issue is the focus of this study.

Concepts are auto-induced in an iterative process [4, 8, 9], shown schematically in Fig. 1. On the left it is shown how a typical sentence from the Travel domain 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. The lexical phraser and the semantic generalizer are described next in some detail.

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* $\rightarrow$ *[New_York]*. 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 [10] to find those lexical entities (referred to as words in the remainder of this paper) that co-occur often. The n phrases with the largest MI measure,

$$MI(w_1, w_2) = p(w_1, w_2) \log \frac{p(w_1, w_2)}{p(w_1)p(w_2)} \quad (1)$$

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 from studying the three smallest corpora that $n = 25$ phrases (or more generally, *chunks*) per iteration were a reasonable number for the small corpora used for three of the four domains in this study. For comparison purposes, the same criterion was used for the much larger WSJ corpus. Fewer than 10 chunks meant that certain commonly occurring phrases, such as *I want*, would not be combined. More than 50 chunks created so many nested chunks, such as *[[go_to]_Newark]*, 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 *Newark* being a member of a <*city_name*> class, where brackets denote a semantic class label) within these large sentence-level chunks.

2.2. Semantic generalization

The next block in Fig. 1, the semantic generalizer, is the primary topic of this paper. 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 generating such groupings (as discussed in the next section) is lexical or semantic similarity of the left of right-hand context for the members of a group. As an example, city names are grouped into the same class because they are used in similar lexical contexts, for example: *I want to fly to* <city_name> *tomorrow*. Ideally, only one semantic merger would be generated each iteration so that the new semantic group could be incorporated into the corpus immediately. To reduce computational complexity five rules were generated per iteration; no qualitative difference was seen, occasionally the order in generating grammar rules was altered. Next, we compare the ability of four different metrics to estimate the degree of similarity between pairs of "words" (phrases or even class labels in the general case) in a bigram context.

2.3. Similarity metrics

The semantic generalizer pairs words or phrases (generated in the preceding lexical phraser module) according to the similarity of their syntactic environments. We consider a candidate word, w, in a word sequence, $\{...\ v^L\ w\ v^R\ ...\}$, with left and right contexts, v^L and v^R. Two probability distributions are calculated, $p^L = p^L(v^L|w)$ and $p^R = p^R(v^R|w)$, for the the left and right bigram contexts respectively. The right-context bigrams are calculated using the usual word order, and the left-context probabilties are calculated with a reversed order training corpus using standard n-gram training tools.

We estimate the similarity of two words, w_1 and w_2, as the sum of the symmetric left and right context-dependent distances [8], giving the total distance

$$D^{LR}(w_1, w_2) = D^L_{12} + D^L_{21} + D^R_{12} + D^R_{21} \qquad (2)$$

where $D^L_{12} = D(p^L_1(v^L|w_1)\ ||\ p^L_2(v^L|w_2))$ is the left-context distance and the DR distance terms are similar, using the right-context probabilities, $p^R(v^R|w)$. The distance, D, used in each term in Eq. 2, is calculated by means of one of four metrics studied in this work. Three of these four metrics are distances: Kullback-Leibler (KL), Information-Radius (IR), and the Manhattan-Norm

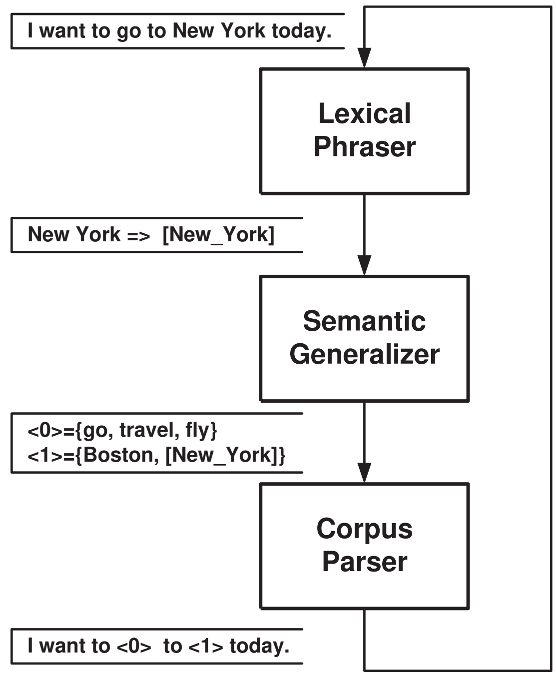

Fig. 1. Schematic view of the iterative procedure used for the auto-induction of semantic classes.

(MN). The fourth, the Vector-Product (VP), is actually a similarity measure.

The KL distance has been commonly used for the auto-induction of classes [4, 8, 9]. However, the KL metric 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 terms, or even just one. This inspired us to investigate three other metrics, all bounded. We compared them by subjectively evaluating the quality of the semantic classes generated by each. The Kullback-Leibler, Information Radius and Manhattan Norm are distance measurements. The fourth, the Vector Product, is a similarity measure.

2.3.1. *Kullback-Leibler distance (KL)*

The Kullback-Leibler (KL) distance is a relative entropy measure [11] of the distance between two distributions, p_1 and p_2. Eq. 3 shows the KL distance between two bigram contexts (left-context) as a means of determining the similarity between two words, w_1 and w_2.

Following Eq. 2, the total, symmetric KL distance is given by $K^{LR}(w_1, w_2) = K^L_{12} + K^L_{21} + K^R_{12} + K^R_{21}$. As a representative example, the left-context dependent KL distance K^L_{12} between two candidate words, w_1 and w_2, is defined over the vocabulary V as

$$K^L_{12} = K(p^L_1(w_1)\ ||\ p^L_2(w_2)) = \\ = \sum_{v \in V} p^L_1(v|w_1)\ \log \frac{p^L_1(v|w_1)}{p^L_2(v|w_2)} \qquad (3)$$

where sum is over all words in the vocabulary, V. In addition, a related problem is that the denominator in the logarithm ratio can be very small in cases where the statistics are poor. 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.

2.3.2. *Information-radius distance (IR)*

The IR distance is similar to the KL distance [6], but is bounded because the denominator for the logarithmic ratio is the average of the two probabilities being considered. The total, symmetric IR distance is given by $I^{LR}(w_1, w_2) = I^L_{12} + I^L_{21} + I^R_{12} + I^R_{21}$. A representative term is,

$$I^L_{12} = \sum_{v \in V} p^L_1(v|w_1) \ \log \left[\frac{p^L_1(v|w_1)}{\frac{1}{2}(p^L_1(v|w_1) + p^L_2(v|w_2))} \right] \qquad (4)$$

with a maximum distance, $\log(2)$ for each of the four terms.

2.3.3. *Manhattan-norm distance (MN)*

The Manhattan-norm (also referred to as the L1 distance) is just the absolute value of the difference of the two distributions and is given by $M^{LR}(w_1, w_2) = M^L_{12} + M^R_{12}$. It has an upper bound of 4, the left and right context sums each being between 0 and 2. The left-context dependent term is

$$M^L_{12} = \sum_{v \in V} \left| \, p^L_1(v|w_1) - p^L_2(v|w_2) \, \right| \qquad (5)$$

where $M_{12} = M_{21}$.

2.3.4. *Vector-product similarity (VP)*

This metric is a similarity measure, rather than a difference measure. This is the vector product of two vectors, each vector being a sequence of bigram probabilities. The total distance is $V^{LR}(w_1, w_2) = V^L_{12} + V^R_{12}$ and each term is bounded by 0 (no similarity) and 1 (identical vectors). The left-context vector product is

$$V^L_{12} = \frac{\displaystyle\sum_{v \in V} p^L_1(v|w_1) \, p^L_2(v|w_2)}{\sqrt{\displaystyle\sum_{v \in V} p^L_1(v|w_1)^2 \, p^L_2(v|w_2)^2}} \qquad (6)$$

and V^{LR} has an upper bound of 2, which is the value for which two words are the closest match.

2.4. The "Stopping Criterion"

The semantic generalizer iteratively creates classes from candidate pairs until the system eventually runs out of candidates. Initially, the merger rules are terminal rules, where words and phrases are merged into new or existing semantic classes; in later iterations, the mergers are predominately merging one class into another. Eventually, all classes are merged into a single "sentence-level" class. Therefore, we need to identify the stopping point, the point where enough merger rules have been generated, identifying semantic classes of interest to a developer of a natural language understanding system. However, we have not yet found a good general stopping criterion; for this work, we chose to evaluate the first m groups provided by the semantic generalizer. In our experience, at least with small semantically homogeneous domains, good results are obtained by stopping after $m = 40$ groups have been generated.

Feature	Carmen	Movie	Travel	WSJ
Sentences	2,416	2,500	1,451	6,920
Words	12,128	16,386	7,811	152,526
Unigrams	433	583	764	13,219
Bigrams	256	368	278	11,441
Trigrams	334	499	240	6,484

Table 1. Statistics for the four domains: Carmen-Sandiego (Carmen), Movie, Travel, and Wall Street Journal (WSJ).

3. EXPERIMENTAL RESULTS

Several methods were used to compare the efficacy of the different metrics. A subjective comparison asked human subjects to evaluate the first 40 groups generated for each metric for each of the four domains in this study. Five naive evaluators labeled each terminal and merger rule for each metric and domain. Each rule could be given a 0 (bad rule), 1 (good rule), or 0.5 (not clear). The agreement between labelers can be determined using the standard kappa-evaluation statistic [12], where $\kappa = 1$ means labelers are in complete agreement and $\kappa = -1$, complete disagreement.[1] In our studies, the average value of κ for all pairs of evaluators, for all metrics and domains, is $\kappa = 0.85$. This value ranged from a minimum of 0.74 for the Carmen domain to a maximum of 0.94 for the WSJ task. This indicates that the labelers are mostly in agreement, although for some domains they agree less than perfectly.

3.1. Four domains used in this study

Table 1 contains the corpus statistics for the four domains used in this study. Three domains used in a previous study [2] are corpora from human-machine conversations: Carmen-Sandiego, a children's computer game; Movie, an information retrieval task; and Travel, an air, hotel, and car reservation system. The first three corpora were small; each corpus contained less than 2500 sentences and fewer than 20,000 words. The fourth domain consists of a subset of 6,920 sentences, and about 150,000 words, from the Wall Street Journal (WSJ) corpora. This WSJ corpus consists of many topics ranging in size from two sentences (about 40 to 50 words) to several dozen sentences (about 1000 words). The WSJ was included in this study to investigate the limitations of automatic concept induction when dealing with a large semantically heterogeneous corpus. In Table 1, the set size for each feature is shown; bigrams and trigrams are only included for extant word sequences. A cutoff threshold of three was used for bigrams and trigrams.

3.2. Some auto-induced classes for the Travel domain

Table 2 shows some of the classes induced in the Travel domain using the VP similarity metric. The classes shown are some of the first 40 classes induced. Most of the classes are reasonably well

[1] The κ statistic is defined as $\kappa = \frac{P(a)-P(c)}{1-P(c)}$ where P(a) is the probability of agreement in labeling and P(c) is the probability of agreement by chance.

Class	Members
<G0>	second, third, sixth, ninth
<G1>	fourteenth, twentieth
<G2>	last, latest, first
<G3>	Boston, Newark, [San_<G21>]
<G4>	[I'd_like], [I_want], [I_need]
<G5>	fourth, fifth
<G14>	airport, seventeenth
<G21>	Antonio, Diego
<G22>	eighteenth, [twenty_<G5>]

Table 2. Sample auto-induced classes for the Travel domain.

defined, matching human judgment. The classes formed predominantly correspond to common travel concepts such as dates, place names, and company names.

Some of the class members are misclassified, such as <G14> $= \{airport, seventeenth\}$. These are word combinations occurring in a similar lexical bigram-context. In the case of <G14>, the most common lexical context was $\{\ldots$ *the* __ */s*$\}$, where */s* is the end of sentence marker. This indicates that the bigram context is sometimes too local to capture semantic similarity.

3.3. A comparison of the four metrics for all four domains

Table 3 shows the number of misclassified elements for each of the four metrics for the four domains studied. Data are for the first 40 classes formed. These forty classes typically included about 120 rules for merging single words and phrases, and about 15 rules for merging two existing classes together. The main exception was the WSJ corpus, which had almost no class-class mergers, due to the large size of the WSJ corpus (see Table 1). There was no appreciable difference observed between a metric's ability to merge individual words and phrases into an existing class, and its ability to merge two existing classes.

The ability of any metric to induce classes was worst for the WSJ corpus, with all four metrics generating semantic classes with more than 76% of the members misclassified. All the metrics use bigram-contexts for probability computation, so phrases of the type, $\{\ldots\ the\ \ w\ \ of\ \ \ldots\}$, classify all words, w, in the same class. In the WSJ corpus, an example is: *The daughter of the firm's founder ...* and *The string of losses ...*. In such cases, $w = (daughter, string)$ is a broad part-of-speech category such as a singular noun. This problem does not occur in the other three domains for three main reasons. Each of the small corpora contains a limited number of query types, limited vocabulary (less than 800 words), and limited number of semantic classes with a precise meaning (such as *<city_name>*).

All four metrics created tightly defined classes for the Movie domain because queries were limited to three types of WH-questions: what, when, where. A typical when-request for this domain was, *When is Lion King playing at Northgate theatre?*

Overall, the bounded metrics perform better than the unbounded KL distance. For example, the IR distance is better than the KL distance because the near-zero probabilities have only a small influence. The poor performance of the VP-similarity may be due to the limited number of extant bigrams.

Domain	IR	KL	MN	VP
Carmen	27.2 %	30.6 %	28.0 %	30.7 %
Movie	8.4	9.6	10.5	12.3
Travel	22.1	25.3	21.7	26.1
WSJ	79.9	88.9	76.1	76.8

Table 3. Percentage of misclassified semantic class members. Metrics used were: Kullback-Leibler (KL), Information-Radius (IR), Manhattan-Norm (MN), and Vector-Product (VP).

4. CONCLUSIONS

We conclude that from the four semantic similarity metrics proposed for auto-inducing semantic classes, the KL and VP metrics perform the worst, while the MN and IR metrics are the best at classifying words and phrases into semantic groups. Good classification results have been demonstrated for three semantically homogeneous domains using these context-based similarity metrics. However, for the large semantically heterogeneous WSJ corpus the bigram-context did not provide adequate information for auto-inducing semantic classes. Further research is needed to investigate other syntactic and lexical features that indicate semantic similarity. More work is also needed for developing the criterion for stopping the iterative semantic generalization process.

5. REFERENCES

[1] A. Pargellis, H.-K. J. Kuo, C.-H. Lee, "Automatic Dialogue Generator Creates User Defined Applications", Proc. of the Sixth European Conf. on Speech Comm. and Tech., Budapest, vol. 3, pp. 1175-1178, 1999.

[2] A. N. Pargellis, E. Fosler-Lussier, C.-H. Lee, A. Potamianos, "Metrics for Measuring Domain Independence of Semantic Classes", Proc. of the Seventh European Conf. on Speech Comm. and Tech., Aalborg, Sept. 2001.

[3] P. F. Brown, et al., "Class-based n-gram models of natural language," Computational Linguistics, vol. 18(4), pp. 467-479, 1992.

[4] M. K. McCandless, J. R. Glass, "Empirical acquisition of word and phrase classes in the ATIS domain," Proc. of the Third European Conf. on Speech Comm. and Tech., Berlin, pp. 981-984, 1993.

[5] A. Gorin, G. Riccardi, J. H. Wright, "How May I Help You?", Speech Communications, vol. 23, pp. 113-127, 1997.

[6] I. Dagan, L. Lee, F. Pereira, "Similarity-Based Methods for Word-Sense Disambiguation", Proc. of the 35th Annual Meeting of the ACL, with EACL 8, 1997

[7] K. Arai, J. H. Wright, G. Riccardi, A. L. Gorin, "Grammar Fragment Acquisition using Syntactic and Semantic Clustering," Proc. Fifth Intl. Conf. on Spoken Lang. Proc., Sydney, vol. 5, pp. 2051-2054, 1998.

[8] K.-C. Siu, H. M. Meng, "Semi-automatic Acquisition of Domain-Specific Semantic Structures," Proc. of the Sixth European Conf. on Speech Comm. and Tech., Budapest, vol. 5, pp. 2039-2042, 1999.

[9] E. Fosler-Lussier, H.-K. J. Kuo, "Using Semantic Class Information for Rapid Development of Language Models within ASR Dialogue Systems," Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Proc., Salt Lake City, 2001.

[10] C. D. Manning and H. Schutze, Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge, 2000.

[11] R. O Duda, P. E. Hart, D. G. Stork, Pattern Classification, John Wiley & Sons, Inc., New York, 2001.

[12] J. Cohen, "A coefficient of agreement for nominal scales." Educational and Pschological Measurement, vol. 20, pp. 307-320, 1960.