ON THE USE OF STRUCTURES IN LANGUAGE MODELS FOR DIALOGUE

Renato De Mori, Yannick Estève, Christian Raymond

LIA-CNRS, University of Avignon
France
renato.demori,yannick.esteve,christian.raymond@lia.univ-avignon.fr

ABSTRACT

The paper describes the combined use of three new language modelling paradigms. They are: generation of plausible trigrams by analogy, explanation-based generation of error-correcting automata, and disambiguation using Semantic Classification Trees. Tangible word error rate reduction is observed by the combined use of these paradigms.

1. INTRODUCTION

Very often dialogue systems are developed without the availability of large corpora for training language models (LM). In spite of this, many sentences follow a limited number of typical patterns. Furthermore, many errors appear in ungrammatical sentences while some errors are due to minimal acoustic variations and correspond to sentences that are syntactically acceptable. Other errors are due to homophones, very frequent in a language like French.

Different types of problems require different solutions. This motivates the approach proposed in this paper which suggests to rescore a trellis of hypothesized words based on different types of LMs obtained by adapting to specific problems some learning methods developed in Artificial Intelligence and Pattern Recognition. These methods are inspired by paradigms known as learning by analogy, explanation-based learning, error correcting parsing and semantic classification. Basic principles of the above mentioned methods are applied to develop new and effective solutions for dialogue applications. Three LMs for hypotheses rescoring will be introduced.

Generation of plausible trigrams by analogy consists in constructing new trigrams not observed in the training set, by replacing words or histories in an observed trigram with other words or histories which have analogy with what they replace. Distances between words and histories are measured in a reduced space obtained with Singular Value Decomposition (SVD) of the matrix of probabilities in which rows correspond to words and columns correspond to histories.

The second method is generation of explanation-based error-correcting automata which starts with the observation of an error, e.g. the absence of a verb. The error and its context are then generalized to obtain a precondition for the application of an error-correcting automaton. The automaton is then obtained by generalizing the corrections.

Homophone or quasi-homophone disambiguation using semantic classification trees (SCT) is the third method which uses sentence patterns detected in the most likely sentence hypothesized in a first recognition phase. The case considered here is that of a confusions between very similar frequently used phrases, which may have similar or even the same phonetic transcription. An SCT generates a correction on a phrase of the hypothesized sentence based on the pattern which applies to the sentence. This is more than using automata, because corrections may involve a selected number of non contiguous words.

The application considered for the experiments described in this paper is a prototype of the vocal server AGS developed at France-Telecom R&D. This is a medium size vocabulary (1000 words) telephone application involving requests of information about telephone services. Most of the recognition problems are related to the expression of a limited number of concepts characterized by a small set of keywords and a much larger number of specification words like geographic locations and job types. As this is a real-world application, many recognition errors are due to background noise, hesitations, correction, erroneous end-point detection, use of out-of-vocabulary (OOV) words. Many errors appear on a fairly limited number of sentence types. Tangible reductions of Word Error Rates (WER) are obtained with different methods because frequent errors are of different type.

2. GENERATION OF PLAUSIBLE TRIGRAMS

If a limited amount of training data is available, many trigrams that would appear more than once in an ideally large

This work was carried out with the support of France Telecom R&D

training corpus have a probability computed with a back-off model. This probability is often much lower than the one that would be computed with a richer training corpus. Furthermore, in many practical applications, the training data available are biased by the fact that they have been collected with a limited number of speakers and in a limited time period. This has the effect that often the probability of certain trigrams is abnormally large.

These considerations suggest that trigram counts have to be adapted. The same adaptation algorithm can be applied to all trigrams or only to certain classes of them for which an algorithm can produce tangible benefits. For certain languages, like French, many errors are due to erroneous recognition of prepositions, articles and other short words appearing in a trigram which include a noun or a verb. For this reason, attention has been focused on trigrams involving the most frequent nouns (e.g. server, number, job, region) and verbs (e.g. call, find, look for, will), as well as geographic names and typical expressions like *toute la France*.

The number of trigram considered is about 200, while the number of trigrams observed in a training set of $70,000$ words is about $10,000$. Each trigram t is represented as $t = hw$, where h represents the history of word h. First of all, let us consider the generation of new trigrams sharing the same history. Let $t' = hv$ be a new trigram derived from t by analogy. The possibility of acquiring new knowledge by analogy was first proposed by [1]. Derivation by analogy can be made by extracting from a very large general corpus all the trigrams $t'' = hx$ such that x belong to the same syntactic class of w. For a limited domain application and for a limited number of histories, generation by analogy can also be done manually. More formally, the generation by analogy of a set of trigrams T' given a set T of observed trigrams is defined by the following logical expression:

$$T' = \left\{ t' = hx \;\middle|\; \begin{array}{l} [(t = hw) \in T] \\ \wedge [POS(x) = POS(w)] \\ \wedge [SEMCOMP(x, w] \end{array} \right\}$$

where $POS(w)$ indicates the syntactic class (the Part Of Speech) of word w. $SEMCOMP(x, w)$ indicates that x and w are semantically compatible words as it will be defined later on.

Let $c(hw)$ be the count of trigram t obtained directly from the training set. Let:

$$c_M(h) = max_{y/hy \in T} c(hy)$$
$$m(h) = argmax_{y/hy \in T} c(hy)$$

The new counts $c'(hz)$ of trigrams inferred by analogy and having history h are recomputed as follows:

$$c'(hz) = \begin{cases} \beta_1 c(hz) & \text{if } c(hz) > \upsilon_1 c_M(h) \\ \alpha_1 c_M(h) e^{-d(x, m(h))} & \text{otherwise} \end{cases}$$

$\alpha_1, \beta_1, \upsilon_1$ are thresholds that can be settled in order to satisfy a condition on the sum of counts for history h. Threshold is set in such a way that, if a count is more than 10% of the maximum count, it should just be multiplied by β_1. If very few of the possible trigrams, analogous because they have the same history, have very high counts, then it is likely that this is the result of a bias in the training set and part of their counts should be redistributed among the analogous trigrams. For words z like prepositions or determiners, it is reasonable to assume that they have the same probability in phrases with history h, leading to the following assumption:

$$c'(hz) = \begin{cases} \beta_1 c(hz) & \text{if } c(hz) > \upsilon_1 c_M(h) \\ \alpha_2 c_M(h) & \text{otherwise} \end{cases}$$

Distance $d(x, m(h))$ is the Euclidian distance between each pair of vectors representing words, computed in a reduced space as proposed in . This distance is also used to define the truth of the $SEMCOMP(x, w)$ predicate which is true when the distance between x and w is lower than a threshold.

Trigrams sharing the whole history or just a word in a history have counts recomputed in a similar way. Other options are possible. An interesting one consists in obtaining a reduced space for counts of word co-occurrence and to compute distances in that space.

3. GENERATION OF EXPLANATION-BASED ERROR-CORRECTING AUTOMATA

3.1. Generation of explanation-based error-correcting automata

Error correcting parser theory ([3]) uses knowledge to augment the rules of a grammar so that a parser using the augmented grammar can parse erroneous sentences. A similar type of rules can be used to invoke regular languages accepted by finite-state automata to be dynamically associated to n-gram LMs in such a way that correct phrases become more likely than hypothesized phrases that are syntactically or semantically incorrect. Details of such a combination are described in [4] and have been extended to context-free languages in [5]. A method is introduced in the following for acquiring error correcting knowledge from examples. It is based on Explanation-Based Learning (EBL), and extracts general knowledge from specific examples ([6]).

In our case, an example is a phrase m in context (a, b) which is incorrect and should be replaced by n. The correction is represented by the statement: $context(a, b) \wedge replace(m, n)$. This logical statement can be seen as a conclusion supported by some other observations, for example that m does not contain a verb while it should and n contains a verb which is semantically and syntactically consistent with context (a, b). This fact can be formally represented by the proof tree in Figure 1. The proof tree in Figure 1 is

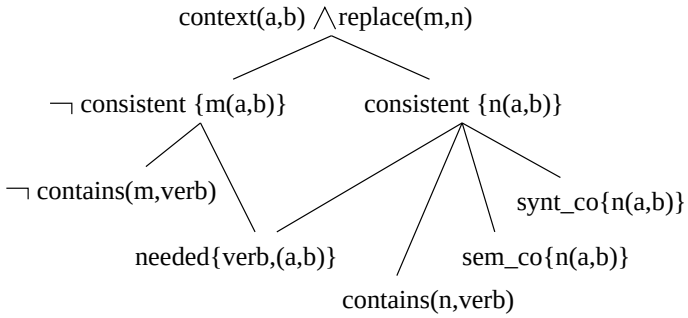


Fig. 1. Proof tree of the example

then generalized by replacing specific phrases m and n with non-terminal variable symbols x and y, leading to a general proof tree for constructing rewriting expressions with which a syntactically and semantically consistent phrase is inserted into in the context (a, b).

This paradigm is applied in practice by considering the most likely hypothesis generated by the recognizer. If the hypothesis contains the context (a, b), the phrase x between a and b is analyzed. If it is inconsistent, then the automaton that generates the regular language of consistent phrases is invoked and used in conjunction with a general n-gram model for rescoring the trellis produced by the recognizer.

A single observation is sufficient for detecting an inconsistence in context (a, b). Then an automaton $A(a, b)$ can be derived manually or by grammatical inference ([3]) on all the phrases obtained from a general large corpus and having context (a, b) and words between a and b belonging to the application lexicon. Eventually, context (a, b) can also be generalized by considering synonyms of words in it. As the training set is limited, interesting cases can be found by simply running the recognizer on the training set.

When specialized stochastic automata are used together with n-gram LMs applicable to entire sentences, the sentence probability is computed by combining the contribution of these different LMs. Let W_{i+1}^{j} be the phrase recognized by the stochastic automata with probability and W_1^i, W_{j+1}^N be respectively the sequence of words preceding and following W_{i+1}^{j}. The sentence probability is given by:

$$P(W_1^i W_{i+1}^j W_{j+1}^N) = P(W_1^i) \left\{ \sum_{k=1}^{K} P(A_k W_{i+1}^j | W_1^j \right\} P(W_{j+1}^N | W_1^i W_{i+1}^j)$$

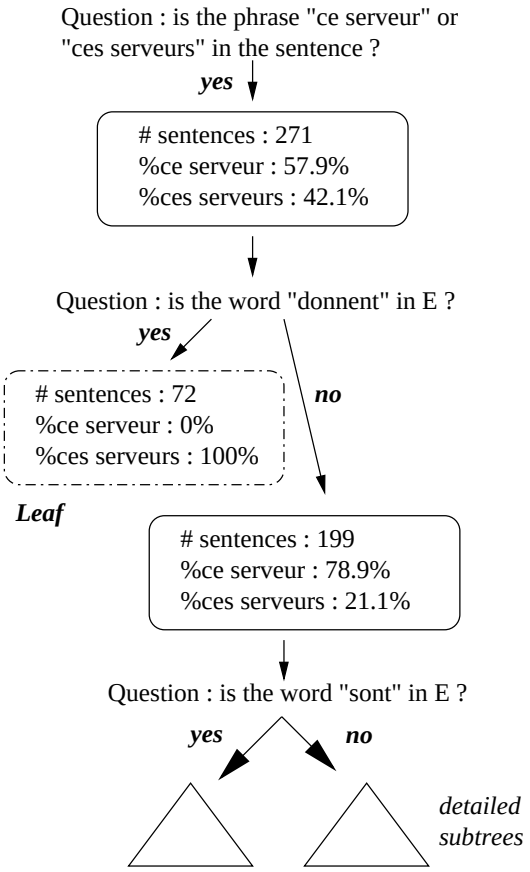
A_k represents an LM which is the n-gram LM if $k = 1$ and a stochastic automaton if $k > 1$. The terms preceding and following the summation are computed with the n-gram LM. Each term of the summation is then decomposed as follows: $P(A_k W_{i+1}^j | W_1^i) = P(W_{i+1}^j | A_k W_1^i) P(A_k | W_1^i)$. The term $P(W_{i+1}^j | A_k W_1^i)$ is computed with the n-gram model if $k = 1$ and is approximated as follows if $k > 1$: $P(W_{i+1}^j | A_k W_1^i) \cong P(W_{i+1}^j | A_k)$. This probability is the one assigned to W_{i+1}^j by the stochastic automaton. The probability $P(A_k | W_1^j)$ is obtained by replacing in the training set a phrase recognized by A_k with the symbol A_k and computing n-gram counts for the sequence $W_1^i A_k$, while $P(A_1 | W_1^i)$ is computed as $1 - \sum_{k=2}^{K} P(A_k | W_1^i)$. In practice, the summation is substituted with maximum and very often it involves only two terms. Furthermore, the contribution of the automaton dominates over that of trigrams because the number of phrases accepted by the automaton is limited.

4. DISAMBIGUISATION USING SEMANTIC CLASSIFICATION TREES

The third proposed paradigm is based on an automatic training method summarized as follows:

- use the training set to identify more frequent and confusable phrases,
- infer, from the training set, sentence patterns for these phrases,
- for each phrase y belonging to a type of phrase confusion, a classification tree is trained using a slightly modified version of the algorithm proposed in [7] to infer sentence patterns for each acceptable sentence of that type. An example of confusion is between phrases $\{ce\ serveur,\ ces\ serveurs\}$ represented by the pattern of lems $pl = (ce * serveur*)$. The classification tree shown in Figure 2 was obtained from a training set of 9842 sentences. Detailed sub-trees at the bottom of the figure are represented by triangles for the sake of simplicity.

Rescoring is based on the following relation, where φ represents a context: $P(W|\varphi) = P_g(B) \times \{P_b(y|BE)P(b|\varphi) + P_g(y|h)P(g|\varphi)\} \times \{P_g(E|yB)P(g|\varphi) + P_b(E|B)P(b|\varphi)\}$



Sentences are represented as S=ByE, where y="ce serveur*", B is the sequence of words before y, and E is the one following y*

"# sentences" represents the number of sentences associated to a node in the training corpus

Fig. 2. Abridged version of a Semantic Classification Tree

The subscript g indicates a general (in our case a trigram) model, the subscript b indicates the new model. In particular, the probability $P_b(y|BE)$ is computed from the leaves of the classification trees. This probability can be used in various ways. In the experiments described in the following, it has been used to decide about corrections to be made on s. The correction which obtained the highest probability after rescoring with the previously proposed methods has been included between B and E to generate the recognition results.

5. EXPERIMENTS AND CONCLUSIONS

Experiments were carried out on the AGS system. Baseline LM was performed on transcribed sentences with a total of $70,000$ words. Test was performed using 1403 sentences uttered by several new speakers.

Results reported in Table 1 show the improvement introduced by the use of analogy and error correction. The use of SCTs was very effective on the sentences on which trees were applicable: experiments on the most frequent homophones and quasi homophones have lead to a WER reduction of 28.9% for the cases where these homophones appeared.

LM	WER
baseline	20.8%
+analogy	20.1%
+error correcting automata	18.7%

Table 1. WER obtained with the the combined use of two new language modelling paradigms

It appears that it is necessary to use several structural models to observe tangible WER reductions.

6. REFERENCES

[1] T.G. Evans (1968): "A program for the solution of a class of geometry analogy intelligence test questions", *In M. Minsly Ed., Semantic Information Processing*, 271-353, MIT Press. Cambridge MA.

[2] D. Janiszek, F. Béchet, R. De Mori (2001): "Data Augmentation and language model adapation", *Proc. ICASSP2001*, Salt Lake City, Utah, USA

[3] K.S. Fu (1982): "Syntactic Pattern Recognition", *Theory and Application*, Prentice Hall

[4] A. Nasr, Y. Estève, F. Béchet, T. Spriet, R. De Mori (1999): "A language model combining n-grams and stochastic finite state automata", *Proc. Eurospeech*99, pp: 2175-2178, Hungary

[5] X. Huang, A. Acero, H.W. Hon (2001): "Spoken Language Processing", Prentice Hall, PTR.

[6] T. Mitchell, R. Keller and S. Kedar-Cabelli (1986): "Explanation-based generalisation: a unified view", *Machine Learning*, 1, 47:80

[7] R. Kuhn and R. De Mori (1995): "The Application of Semantic Classification Trees to Natural Language Understanding", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol PAMI-17, no. 5, 449-460