(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 488 338 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
21.04.2010 Bulletin 2010/16

(21) Application number: **03716920.8**

(22) Date of filing: **27.03.2003**

(51) Int Cl.:
G06F 17/28 $^{(2006.01)}$

(86) International application number:
PCT/US2003/009771

(87) International publication number:
WO 2003/083710 (09.10.2003 Gazette 2003/41)

(54) PHRASE-BASED JOINT PROBABILITY MODEL FOR STATISTICAL MACHINE TRANSLATION

PHRASENBASIERTES GEMEINES WAHRSCHEINLICHKEITSMODELL ZUR STATISTISCHEN MASCHINELLEN ÜBERSETZUNG

MODELE DE PROBABILITE JOINTE PHRASE A PHRASE POUR LA TRADUCTION AUTOMATIQUE DE STATISTIQUES

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT RO SE SI SK TR

(30) Priority: **27.03.2002 US 368450 P**

(43) Date of publication of application:
22.12.2004 Bulletin 2004/52

(73) Proprietor: **UNIVERSITY OF SOUTHERN CALIFORNIA**
Los Angeles, CA 90007-4344 (US)

(72) Inventors:
- **MARCU, Daniel**
  Hermosa Beach, CA 90254 (US)
- **KNIGHT, Kevin**
  Hermosa Beach, CA 90245 (US)
- **WONG, William**
  Mission Viejo, CA 92692 (US)
- **KOEHN, Philipp**
  Venice, CA 90291 (US)

(74) Representative: **Lloyd, Patrick Alexander Desmond**
Reddie & Grose
16 Theobalds Road
London
WC1X 8PL (GB)

(56) References cited:
- **DANIEL MARCU AND WILLIAM WONG : "A PHRASE-BASED, JOINT PROBABILITY MODEL FOR STATISTICAL MACHINE TRANSLATION" PROCEEDINGS OF THE CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING (EMNLP-2002), [Online] 6 - 7 July 2002, XP002280146 Philadelphia, PA, USA Retrieved from the Internet: <URL:http: //www.isi.edu/~marcu/papers/join tmt2002.pdf> [retrieved on 2004-05-12]**
- **DAN MELAMED: "Empirical Methods for Exploiting Parallel Texts" 1 March 2001 (2001-03-01) , THE MIT PRESS XP002280151 page 81 -page 121**
- **FRANZ JOSEF OCH, CHRISTOPH TILLMANN, HERMANN NEY: "Improved Alignment Models for Statistical Machine Translation." PROC. OF THE JOINT CONF. OF EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING AND VERY LARGE CORPORA, [Online] June 1999 (1999-06), pages 20-28, XP002280147 University of Maryland, College Park, MD, USA Retrieved from the Internet: <URL:http: //acl.ldc.upenn.edu/W/W99/W99-06 04.pdf> [retrieved on 2004-05-12]**
- **DANIEL MARCU: "Towards a Unified Approach to Memory- and Statistical-Based Machine Translation" PROCEEDINGS OF ACL-2001, [Online] 5 July 2001 (2001-07-05), XP002280148 Toulouse, France Retrieved from the Internet: <URL:http://www.isi.edu/~marcu/papers/tran smem-acl01.pdf> [retrieved on 2004-05-12]**

Printed by Jouve, 75001 PARIS (FR)

(Cont. next page)

- BROWN P F ET AL: "THE MATHEMATICS OF STATISTICAL MACHINE TRANSLATION: PARAMETER ESTIMATION" COMPUTATIONAL LINGUISTICS, CAMBRIDGE, MA, US, vol. 19, no. 2, June 1993 (1993-06), pages 263-311, XP008022787
- CHRISTOPH TILLMANN AND FEI XIA: " A Phrase-Based Unigram Model for Statistical Machine Translation" HLT-NAACL 2003, [Online] 27 May 2003 (2003-05-27) - 1 June 2003 (2003-06-01), XP002280149 Edmonton, Canada Retrieved from the Internet: <URL:http: //acl.ldc.upenn.edu/N/N03/N03-20 36.pdf> [retrieved on 2004-05-12]
- S. VOGEL, Y. ZHANG, F. HUANG, A. VENUGOPAL, B. ZHAO, A. TRIBBLE, M. ECK, A. WAIBEL: "The CMU Statistical Machine Translation System" MACHINE TRANSLATION SUMMIT IX, [Online] 23 - 27 September 2003, XP002280150 New Orleans, Louisiana, USA Retrieved from the Internet: <URL:http: //www.amtaweb.org/summit/MTSummi t/ FinalPapers/105-vogel-final.pdf> [retrieved on 2004-05-12]

Description

ORIGIN OF INVENTION

[0001] The research and development described in this application were supported by DARPA-ITO under grant number N66001-00-1-9814 and by NSF-STTR grant 0128379. The U.S. Government may have certain rights in the claimed inventions.

BACKGROUND

[0002] Most of the noisy-channel-based models used in statistical machine translation (MT) are conditional probability models. In the noisy-channel framework, each source sentence "e" in a parallel corpus is assumed to "generate" a target sentence "f" by means of a stochastic process, whose parameters are estimated using traditional Expectation Maximum (EM) techniques. The generative model explains how source words are mapped into target words and how target words are re-ordered to yield well formed target sentences. A variety of methods are used to account for the re-ordering of target words, including methods using word-based, template based, and syntax-based models (to name just a few). Although, these models use different generative processes to explain how translated words are re-ordered in a target language, at the lexical level these models assume that source words are individually translate into target words.

[0003] The paper "Towards a Unified Approach to Memory - and Statistical-Based Machine Translation" by Daniel Marcu, Proceedings of ACL-2001, discusses a known statistical machine translation method including a word-based joint probability model that is subsequently trained to develop a translation memory for phrase to phrase correspondence. Other references of interest as background include "The Mathematics of Statistical Machine Translation: Parameter Estimation" by P Brown et al, Computational Linguistics, Cambridge, MA; "Empirical Methods for Exploiting Parallel Texts" by Dan Melamed, The MIT Press. "Improved Alignment Models for Statistical Machine Translation", by Franz Josef Och et al, Procedures of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora went beyond the original statistical machine translation models by allowing multi-word units or phrases to be translated.

SUMMARY

[0004] The invention is defined in the independent claims to which reference should now be made. Advantageous features are set out in the dependent claims.

[0005] A machine translation (MT) system may develop probabilistic phrase-to-phrase translation lexicons using one or more bilingual corpora. For example, translation lexicons may be developed using a joint probability method, a word-to-word conditional method, or other method.

[0006] The MT system may translate one or more sentences (or sentence fragments) using translation lexicons. For example, the MT system may use a greedy method, a method using a beam stack decoder, or other method to decode sentences.

[0007] In implementations in which translation lexicons are developed using a phrase-based joint probability model, source and target language sentences may be generated simultaneously. The system may utilize the joint probability model for both source-to-target and target-to-source translation applications.

[0008] In embodiments using a word-to-word conditional method, the model may learn phrase-to-phrase alignments from word-to-word alignments generated by a word-to-word statistical MT system.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Figure 1 is a block diagram of a machine translation (MT) system including a phrase-based joint probability translation model.

[0010] Figure 2 shows alignments and probability distributions generated by the phrase-based joint probability model.

[0011] Figure 3 is a flowchart describing a training algorithm for the phrase-based joint probability model.

[0012] Figure 4 is shows an example of phrase-based greedy decoding.

[0013] Figure 5 is a flowchart describing a phrase-based decoding algorithm according to an example.

[0014] Figure 6 shows pseudo code describing the phrase-based decoding algorithm.

[0015] Figure 7 is a diagram showing generation of an arc between hypotheses.

[0016] Figure 8 is a graph showing the effect of phrase length on performance.

[0017] Figure 9 shows an example estimation of a lexical weight.

[0018] Figure 10 is a graph showing the effect of lexical weighting on performance.

[0019] Figure 11 is a graph comparing the performance of different heuristics.

DETAILED DESCRIPTION

[0020] Systems and techniques of the current disclosure may be used to provide more efficient and accurate machine translation (MT). In some implementations, the MT system may develop phrase-to-phrase probabilistic translation lexicons. The probabilistic translation lexicons may be automatically learned from bilingual corpora using, for example, joint probability models or word-to-word conditional models.

[0021] These translation lexicons may then be used to translate new sentences. That is, the translation lexicons may be used to translate sentences not included in the corpora used to train the MT system. Systems and techniques for translation include a greedy method, a method using a beam stack decoder, or other methods.

[0022] Figure 1 shows a machine translation (MT) system 100 including a translation model 105 and a decoder 110. Translation model 105 may include translation lexicons that may be learned from bilingual corpora. Translation model 105 may assume that lexical correspondences can be established at the word level and the phrase level as well. Decoder 110 may use the translation lexicons to provide a translated sentence,based on an input sentence.

[0023] Phrase-to-phrase translation lexicon development

[0024] According to some embodiments, model 105 may be trained according to a joint probability model. That is, model 105 may develop a translation lexicon automatically using a parallel corpus 115 including parallel source and target language strings. Model 105 does not try to capture how source sentences can be mapped into target sentences, but rather generates source and target sentences simultaneously. In other words, the translation model is a joint probability model that can be easily marginalized in order to yield conditional probability models for both source-to-target and target-to source machine translation applications.

[0025] In an embodiment, model 105 may generate sentence pairs using the following stochastic process:

[0026] 1. Generate a bag of concepts C.

[0027] 2. For each concept $c_i \in C$, generate a pair of phrases $(\vec{e}_i, \vec{f}_i)$, according to the distribution $t(\vec{e}_i, \vec{f}_i)$, where $\vec{e}_i$ and $\vec{f}_i$ each contain at least one word.

[0028] 3. Order the phrases generated in each language so as to create two linear sequences of phrases; sequences correspond to the sentence pairs in a bilingual corpus.

[0029] For simplicity, it is assumed that the bag of concepts and the ordering of the generated phrases are modeled by uniform distributions. It is also assumed that $c_i = (\vec{e}_i, \vec{f}_i)$. Under these assumptions, it follows that the probability of generating a sentence pair (E, F) using concepts $c_i \in C$ is given by the product of all phrase-to-phrase translation probabilities, $\prod_{C_i \in C} (\vec{e}_i, \vec{f}_i)$ that yield bags of phrases that can be ordered linearly so as to obtain the sentences E and F.

[0030] Figure 2 illustrates an example. The sentence pair "a b c"--"x y" can be generated using two concepts, ("a b": "y") and (" c" : "x"), or one concept, (" a b c": "x y"), because in both cases the phrases in each language can be arranged in a sequence that would yield the original sentence pair. However, the same sentence pair cannot be generated using the concepts (" a b": "y") and ("c" : "y") because the sequence "x y" cannot be recreated from the two phrases "y" and "y". Similarly, the pair can not be generated using concepts ("a c" : "x") and (" b": "y") because the sequence "a b c" cannot be created by concatenating the phrases "a c" and "b".

[0031] The set of concepts C can be linearized into a sentence pair (E, F) if E and F can be obtained by permuting the phrases $\vec{e}_i$ and $\vec{f}_i$ that characterize all concepts $c_i \in C$. We denote this property using the predicate L(E, F, C). Under this model, the probability of a given sentence pair (E, F) can then be obtained by summing up over all possible ways of generating bags of concepts $c_i \in C$ that can be linearized to (E, F).

[0032]

$$p(E,F) = \sum_{c = c \mid L(E,F,C)} \prod_{c_i \in C} t(\vec{e}_i, \vec{f}_i)$$

[0033] The model described above ("Model 1") has been found to produce fairly good alignments. However, this model may be unsuited for translating unseen sentences, as it imposes no constraints on the ordering of the phrases associated with a given concept. In order to account for this, a modified model ("Model 2") was developed to account for distortions. The generative story of the model is this:

[0034] 1. Generate a bag of concepts C.

[0035] 2. Initialize E and F to empty sequences e.

[0036] 3. Randomly take a concept $c_i \in C$ and generate a pair of phrases $(\vec{e}_i, \vec{f}_i)$, according to the distribution $t(\vec{e}_i, \vec{f}_i)$, where $\vec{e}_i$ and $\vec{f}_i$ each contain at least one word. Remove then c_i from C.

[0037] 4. Append phrase $\vec{f}_i$ at the end of F. Let k be the start position of $\vec{f}_i$ in F.

[0038] 5. Insert phrase $\vec{e}_i$ at position 1 in E provided that no other phrase occupies any of the positions 1 and 1+$|\vec{e}_i|$, where $|\vec{e}_i|$ gives the length of the phrase $\vec{e}_i$. The system hence create the alignment between the two phrases $\vec{e}_i$ and $\vec{f}_i$ with probability

$$\prod_{p=k}^{k+|\vec{f}|} d(p,(l+|\vec{e}_i|)/2),$$

where d(i, j) is a position-based distortion distribution.

[0039] 6. Repeat steps 3 to 5 until C is empty.

[0040] In this model, the probability to generate a sentence pair (E, F) is given by the following formula:

$$p(E,F)=\sum_{C\in c|L(E,F,C)}\prod_{c_i\in C}\left[t(e_i,f_{i)}\times\prod_{k=1}^{|\vec{f}_i|}d(pos(f_i^{\,k}),pos_{cm}(\vec{e_i}))\right]$$

where pos ($\vec{f}_i^{\,k}$) denotes the position of word k of phrase $\vec{f}_i$ in sentence F and $\mathrm{pos}_{\mathrm{cm}}(\vec{e}_i^{\,k})$ denotes the position in sentence E of the center of mass of phrase e_i.

[0041] Training the models described above may be computationally challenging. Since there is an exponential number of alignments that can generate a sentence pair (E, F), the Expectation Maximum (EM) training algorithm cannot be applied exhaustively. Figure 3 is a flowchart describing a training algorithm 300 for the phrase-based joint probability model which takes this problem into account.

[0042] The system determines high-frequency n-grams in E and F (block 305). If one assumes from the outset that any phrases $\vec{e}_i \in E^*$ and $\vec{f}_i \in F^*$ can be generated from a concept c_i, one would need a supercomputer in order to store in the memory a table that models the $t(\vec{e}_i, \vec{f}_i)$ distribution. Since the system doesn't have access to computers with unlimited memory, the system initially learns t distribution entries only for the phrases that occur often in the corpus and for unigrams. Then, through smoothing, the system learns t distribution entries for the phrases that occur rarely as well. In order to be considered in the next step of the algorithm, a phrase has to occur at least five times in the corpus.

[0043] The next step is to initialize the t-distribution table (block 310). Before the EM training procedure starts, one has no idea what word/phrase pairs are likely to share the same meaning. In other words, all alignments that can generate a sentence pair (E, F) can be assumed to have the same probability. Under these conditions, the evidence that a sentence pair (E, F) contributes to the fact that $(\vec{e}_i, \vec{f}_i)$ are generated by the same concept c_i is given by the number of alignments that can be built between (E,F) that have a concept c_i that is linked to phrase $\vec{e}_i$ in sentence E and phrase $\vec{f}_i$ in sentence F divided by the total number of alignments that can be built between the two sentences. Both these numbers can be easily approximated.

[0044] Given a sentence E of 1 words, there are S(l, k) ways in which the l words can be partitioned into k non-empty sets/concepts, where S(l, k) is the Stirling number of second kind.

[0045]

$$S(l,k)=\frac{1}{k!}\sum_{i=0}^{k-1}(-1)^i\binom{k}{i}(k-i)^n$$

[0046] There are also S(m, k) ways in which the m words of a sentence F can be partitioned into k non-empty sets. Given that any words in E can be mapped to any words in F, it follows that there are $\sum_{k=1}^{\min(l,m)} k!S(s(l,k)S(m,k)$

alignments that can be built between two sentences (E, F) of lengths 1 and m, respectively. When a concept ci generates two phrases ($\vec{e}_i$, $\vec{f}_i$) of length a and b, respectively, there are only 1-a and m-b words left to link. Hence, in the absence of any other information, the probability that phrases $\vec{e}_i$ and $\vec{f}_i$ are generated by the same concept c_i is given by the following formula:

$$\frac{\sum_{k=1}^{\min(l-a,m-b)} k!S(s(l-a,k)S(m-b,k))}{\sum_{k=1}^{\min(l,m)} k!S(s(l,k)S(m,k))}$$

[0047] Note that the fractional counts returned by the formula are only an approximation of the t distribution the system is interested in because the Stirling numbers of the second kind do not impose any on the words that are associated with a given concept be consecutive. However, since the formula overestimates the numerator and denominator equally, the approximation works well in practice.

[0048] In the second step of the algorithm, the system applies the formula to collect fractional counts for all unigram and high-frequency n-gram pairs in the Cartesian product defined over the phrases in each sentence pair (E, F) in a corpus. The system sums over all these t-counts and normalizes to obtain an initial joint distribution t. This step amounts to running the EM algorithm for one step over all possible alignments in the corpus.

[0049] In the third step of the algorithm., the system performs EM training on Viterbi alignments (block 315). Given a non-uniform t distribution, phrase-to-phrase alignments have different weights and there are no other tricks one can apply to collect fractional counts over all possible alignments in polynomial time. Starting with block 315 of the algorithm in Figure 3, for each sentence pair in a corpus, the system greedily produces an initial alignment by linking together phrases so as to create concepts that have high t probabilities. The system then hillclimbs towards the Viterbi alignment of highest probability by breaking and merging concepts, swapping words between concepts, and moving words across concepts. The system computes the probabilities associated with all the alignments the system generated during the hillclimbing process and collects t counts over all concepts in these alignments.

[0050] The system applies this Viterbi-based EM training procedure for a few iterations. The first iterations estimate the alignment probabilities using Model 1. The rest of the iterations estimate the alignment probabilities using Model 2.

[0051] During training, the system applies smoothing so the system can associate non-zero values to phrase-pairs that do not occur often in the corpus.

[0052] At the end of the training procedure, the system takes marginals on the joint probability distributions t and d (block 320). This yields conditional probability distributions t($\vec{e}_i$, $\vec{f}_i$) and d (posF l posE), which the system uses for decoding.

[0053] When the system runs the training procedure in Figure 3 on the corpus in Figure 2, after four Model 1 iterations the system obtain the alignments 205 and the joint and conditional probability distributions 210. At prima facie, the Viterbi alignment for the first sentence pair may appear incorrect because humans have a natural tendency to build alignments between the smallest phrases possible. However, note that the choice made by our model is quite reasonable. After all, in the absence of additional information, the model can either assume that "a" and "y" mean the same thing or that phrases "a b c" and "x y" mean the same thing. The model chose to give more weight to the second hypothesis, while preserving some probability mass for the first one.

[0054] Also note that although the joint distribution puts the second hypothesis at an advantage, the conditional distribution does not. The conditional distribution 210 is consistent with our intuitions that tell us that it is reasonable both to translate "a b c" into "x y", as well as "a" into "y". The conditional distribution mirrors our intuitions.

[0055] In an alternative arrangement, a system such as system 100 of FIG. 1 may learn phrase-to-phrase translations from word-to-word alignments. That is, a model such as model 105 may develop a phrase translation lexicon by expanding word-to-word translation lexicons learned by word-to-word models. The phrase translation model is based on the noisy channel model. The system uses Bayes rule to reformulate the translation probability for translating a foreign sentence f into English e as

$$\mathrm{argmax}_e p(e|f) = \mathrm{argmax}_e p(f|e)p(e)$$

[0056] This allows for a language model p(e)and a separate translation model F(flle).

[0057] During decoding (i.e., translation), the foreign input sentence f is segmented into a sequence of I phrases $\bar{f}_1^I$. The system assumes a uniform probability distribution over all possible segmentations.

[0058] Each foreign phrase $\bar{f}_1$ in $\bar{f}_1^I$ is translated into an English phrase $\bar{e}_i$. The English phrases may be re ordered.

Phrase translation is modeled by a probability distribution $\phi(\bar{f}_i, \mathrm{l}\bar{e}_i)$. Due to the Bayes rule, the translation direction is inverted from a modeling standpoint.

[0059] Reordering of the English output phrases is modeled by a relative distortion probability distribution $d(a_i - b_{i-1})$, where a_i denotes the start position of the foreign phrase that was translated into the itch English phrase, and b_{i-1} denotes the end position of the foreign phrase translated into the(i - 1)th English phrase.

[0060] The distortion probability distribution d(·) may be trained using a joint probability model, such as that described in connection with the previous described arrangement. Alternatively, the system could also use a simpler distortion model $d(a_i - b_{i-1}) = \alpha^{\mathrm{l}a_i - b_{i-1} - \mathrm{il}}$ with an appropriate value for the parameter α.

[0061] In order to calibrate the output length, the system introduces a factor ω for each generated English word in addition to the trigram language model p_{LM}. This is a simple means to optimize performance. Usually, this factor is larger than 1, biasing longer output.

[0062] In summary, the best English output sentence e_{best} given a foreign input sentence f according to the model is

$$\begin{aligned} E_{best} &= \mathrm{argmax}_e\, p(e|f) \\ &= \mathrm{argmax}_e\, p(f|e)\, p_{LM}(e)\, \omega^{length(e)} \end{aligned}$$

where p(fle) is decomposed into

$$p(\bar{f}_1^I | \bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i | \bar{e}_i) d(a_i - b_{i-1})$$

[0063] The Giza++ toolkit was developed to train word-based translation models from parallel corpora. As a byproduct, it generates word alignments for this data. The system may improve this alignment with a number of heuristics. The system collects all aligned phrase pairs that are consistent with the word alignment. The words in a legal phrase pair are only aligned to each other, and not to words outside. Given the collected phrase pairs, the system estimates the phrase translation probability distribution by relative frequency:

$$\phi(\bar{f} | \bar{e}) = \frac{count(\bar{f} | \bar{e})}{\sum_{\bar{f}} count(\bar{f} | \bar{e})}$$

[0064] In some arrangements, smoothing may be performed.

[0065] If the system collects all phrase pairs that are consistent with word alignments, this includes many non-intuitive phrases. For instance, translations for phrases such as "house the " may be learned. Intuitively the system would be inclined to believe that such phrases do not help. Restricting possible phrases to syntactically motivated phrases could filter out such non-intuitive pairs.

[0066] Another motivation to evaluate the performance of a phrase translation model that contains only syntactic phrases comes from recent efforts to built syntactic translation models. In these models, reordering of words is restricted to reordering of constituents in well-formed syntactic parse trees. When augmenting such models with phrase translations, typically only translation of phrases that span entire syntactic subtrees is possible. It is important to know if this is a helpful or harmful restriction.

[0067] The system may define a syntactic phrase as a word sequence that is covered by a single subtree in a syntactic parse tree. We collect syntactic phrase pairs as follows: the system word-aligns a parallel corpus, as described above. The system then parses both sides of the corpus with syntactic parsers. For all phrase pairs that are consistent with the word alignment, the system additionally checks if both phrases are subtrees in the parse trees. Only these phrases are included in the model. Hence, the syntactically motivated phrase pairs learned are a subset of the phrase pairs learned without knowledge of syntax. The phrase translation probability distribution may be estimated by relative frequency.

[0068] Figure 8 displays results from experiments with different maximum phrase lengths. All phrases consistent with the word alignment (AP) were used. As shown in Figure 8, limiting the length to a maximum of only three words per phrase already achieves top performance. Learning longer phrases does not yield any improvement. Reducing the limit to only two, however, is detrimental. Allowing for longer phrases increases the phrase translation table size. The increase is almost linear with the maximum length limit. Still, none of these model sizes caused memory problems.

[0069] The system may validate the quality of a phrase translation pair by checking how well its words translate to each other. For this, a lexical translation probability distribution w(fle) may be used. The distribution may be estimated by relative frequency from the same word alignments as the phrase model

$$w(f\mid e)=\frac{count(f,e)}{\sum_{f'}count(f',e)}$$

[0070] A special English NULL token may be added to each English sentence and aligned to each unaligned foreign word.

[0071] Given a phrase pair $\bar{f},\bar{e}$ and a word alignment a between the foreign word positions I - 1,...,n and the English word positions j = 0, 1, ...,m, the system computes the lexical weight p_w by

$$p_w(\bar{f}\mid\bar{e},a)=\prod_{i=1}^{n}\frac{1}{|\{j\mid(i,j)\in a\}|}\sum_{\forall(i,j)\in a}w(f_i\mid e_i)$$

[0072] Figure 9 shows an example.

[0073] If there are multiple alignments a for a phrase pair $(\bar{f},\bar{e})$, the system may use the alignment with the highest lexical weight:

$$p_w(\bar{f}\mid\bar{e})=\max_a p_w(\bar{f}\mid\bar{e},a)$$

[0074] The system may use the lexical weight p_w during translation as an additional factor. This means that the model p(fle) is extended to

$$p(\bar{f}_1^I\mid\bar{e}_1^I)=\prod_{i=1}^{I}\phi(\bar{f}_i\mid\bar{e}_i)d(a_i-b_{i-1})p_w(\bar{f}_i\mid\bar{e}_i,a)^{\lambda}$$

[0075] The parameter λ defines the strength of the lexical weight p_w. Good values for this parameter are around 0.25.

[0076] Figure 10 shows the impact of lexical weighting on machine translation performance. In our experiments, the system achieved improvements of up to 0.01 on the BLEU score scale.

[0077] Phrase translation with a lexical weight is a special case of the alignment template model with one word class for each word. The simplification performed by the system has the advantage that the lexical weights can be factored into the phrase translation table beforehand, speeding up decoding. In contrast to the beam search decoder for the alignment template model, the decoding method described in connection with Figures 5 and 6, are able to search all possible phrase segmentation of the input sentence, instead of choosing one segmentation before decoding.

[0078] In the experiment, the system learned phrase pairs from word alignments generated by Giza++. The IBM Models that this toolkit implements only allow at most one English word to be aligned with a foreign word. The system remedies this problem with a heuristic approach.

[0079] First, the system aligns a parallel corpus bidirectionally, i.e., foreign to English and English to foreign. This gives two word alignments that the system tries to reconcile. If the system intersects the two alignments, the system gets a high-precision alignment of high-confidence alignment points. If the system takes the union of the two alignments, the system gets a high-recall alignment with additional alignment points.

[0080] The space between intersection and union may be expansion heuristics that start with the intersection and add additional alignment points. The decision which points to add may depend on a number of criteria, e.g., which alignment does the potential alignment point exist (Foreign-English or English-Foreign), whether the potential point neighbor already established points, whether "neighboring" means directly adjacent (block-distance), or also diagonally adjacent whether the English or the foreign word that the potential point connects are unaligned so far, and if both are unaligned and the lexical probability for the potential point. ,

[0081] The system starts with intersection of the two word alignments. The system only adds new alignment points that exist in the union of two word alignments. The system also always requires that a new alignment point connects at least one previously unaligned word.

[0082] First, the system expands to only directly adjacent alignment points. The system checks for potential points starting from the top right comer of the alignment matrix, checking for alignment points for the first English word, then continues with alignment points for the second English word, and so on. This is done iteratively until no more alignment point can be added. In a final step, the system adds non-adjacent alignment points, with otherwise the same requirements.

[0083] Figure 11 shows the performance of this heuristic (base) compared against the two mono-directional alignments (e2f, f2e) and their union (union). The figure also contains two modifications of the base heuristic: In the first (diag) the system also permit diagonal neighborhood in the iterative expansion stage. In a variation of this (diag-and), the system requires in the final step that both words are unaligned.

[0084] The ranking of these different methods varies for different training corpus sizes. For instance, the alignment f2e starts out second to worst for the 10,000 sentence pair corpus, but ultimately is competitive with the best method at 320,000 sentence pairs. The base heuristic is initially the best, but then drops off. The discrepancy between the best and the worst method is quite large, about 0.2 BLEU (an IBM scoring system), for almost all training corpus sizes, albeit not always significantly.

[0085] Decoding

[0086] The phrase-based decoder in some embodiments may employ a beam search algorithm. The English output is generated left to right in form of partial translations (or hypotheses).

[0087] The system may begin the search of possible translations in an initial state where no foreign input words are translated and no English output words have been generated. New states may be created by extending the English output with a phrasal translation that covers some of the foreign input words not yet translated. The current cost of the new state is the cost of the original state multiplied with the translation, distortion and language model costs of the added phrasal translation.

[0088] Each search space (hypothesis) is represented by (a) a back link to the best previous state, (b) the foreign words covered so far, (c) the last two English words generated (needed for computing future language model costs), (d) the end of the last foreign phrase covered (needed for computing future distortion costs), (e) the last added English phrase (needed for reading the translation from a path of hypotheses), (f) the cost so far, and (g) the estimate of the future cost.

[0089] Final states in the search are hypotheses that cover all foreign words. Among these the hypothesis with the lowest cost is selected as best translation.

[0090] Two hypotheses can be merged, if they agree in (a) the foreign words covered so far, (b) the last two English words generated, and (c) the end of the last foreign phrase covered.

[0091] If there are two paths that lead to two hypotheses that agree in these properties, the system keeps the cheaper hypothesis, e.g., the one with less cost so far. The other hypothesis cannot be part of the path to the best translation, and the system can safely discard it. Note that the inferior hypothesis can be part of the path to the second best translation.

[0092] Figure 5 is a flowchart describing a phrase-based decoding operation 500 according to an example. An algorithm describing the operation is shown in Figure 6. The system may start with an initial empty hypothesis. A new hypothesis is then expanded from an existing hypothesis by the translation of a phrase. A sequence of untranslated foreign words and a possible English phrase translation for them is selected (block 505). The English phrase is attached to the existing English output sequence (block 510). Then the foreign words are marked as translated and the probability cost of the hypothesis is updated (block 515). The cheapest (highest probability) final hypothesis with no untranslated foreign words is the output of the search (block 520).

[0093] The hypotheses are stored in stacks. The stack s_m contains all hypotheses in which m foreign words have been translated. The system may recombine search hypotheses. While this reduces the number of hypotheses stored in each stack somewhat, stack size is exponential with respect to input sentence length. This makes an exhaustive search impractical.

[0094] Thus, the system prunes out weak hypotheses based on the cost they incurred so far and a future cost estimate. For each stack, the system only keeps a beam of the best n hypotheses. Since the future cost estimate is not perfect, this leads to search errors. Our future cost estimate takes into account the estimated phrase translation cost, but not the expected distortion cost.

[0095] For each possible phrase translation anywhere in the sentence (referred to as a "translation option"), the system multiplies its phrase translation probability with the language model probability for the generated English phrase. As language model probability, the system may use the unigram probability for the first word, the bigram probability for the second, and the trigram probability for all following words.

[0096] Given the costs for the translation options, the system can compute the estimated future cost for any sequence of consecutive foreign words by dynamic programming. Note that this is only possible, since the system ignores distortion costs. Since there are only n(n+1)/2 such sequences for a foreign input sentence of length n, the system can pre-compute these cost estimates beforehand and store them in a table.

[0097] During translation, future costs for uncovered foreign words can be quickly computed by consulting this table. If a hypothesis has broken sequences of untranslated foreign words, the system look up the cost for each sequence and take the product of their costs.

[0098] The space of hypotheses generated during the beam search forms a lattice of paths, each representing a translation, for which a translation score can be easily computed. Extracting the n-best paths from such a lattice is a well-studied problem.

[0099] Paths branch out, when there are multiple translation options for a hypothesis from which multiple new hypotheses can be derived. Paths join, when hypotheses are merged. As described above, the system may discard a hypothesis if it agrees with a lower-cost hypothesis with some of the same properties. In order to keep the information about merging paths, the system keeps a record of such mergings that contains identifier of the previous hypothesis, identifier of the lower-cost hypothesis, and cost from the previous to higher-cost hypothesis.

[0100] Figure 7 gives an example for the generation of such an arc. In this case, the hypotheses 2 and 4 are equivalent in respect to the heuristic search, as detailed above. Hence, hypothesis 4 is deleted. But to retain the information about the path leading from hypothesis 3 to 2, the system stores a record of this arc 705. The arc also contains the cost added from hypothesis 3 to 4. Note that the cost from hypothesis 1 to hypothesis 2 does not have to be stored, since it can be recomputed from the hypothesis data structures.

[0101] The beam size, e.g., the maximum number of hypotheses in each stack, may be fixed to a certain number. The number of translation options is linear with the sentence length. Hence, the time complexity of the beam search is quadratic with sentence length, and linear with the beam size.

[0102] Since the beam size limits the search space and therefore search quality, the system has to find the proper trade-off between speed (low beam size) and performance (high beam size). In experiments, a beam size of only 100 proved to be sufficient. With larger beams sizes, only a few sentences were translated differently. The decoder translated 1755 sentence of length 5 to 15 words in about 10 minutes on a 2 GHz Linux® system. The system achieved fast decoding, while ensuring high quality.

[0103] In some embodiments, a decoder such as decoder 110 of FIG. 1 may implement a greedy procedure. Given a foreign sentence F, the decoder first produces gloss of it by selecting phrases in E* that the probability p(E, F). The decoder then iteratively hillclimb by modifying E and the alignment between E and F so as to maximize the formula p (E) p (F I E). The decoder hillclimbs by modifying an existing alignment/translation through a set of operations that modify locally the alignment/translation built until a given time. These operations replace the English side of an alignment with phrases of different probabilities, merge and break existing concepts, and swap words across concepts. The probability p(E) is computed using a simple trigram language model. The language model is estimated at the word (not phrase) level. Figure 3 shows the steps taken by the decoder in order to find the translation of sentence "je vais me arrêter là." Each intermediate translation 405 in Figure 4 is preceded by its probability 410 and succeeded by the operation that changes it to yield a translation of higher probability.

[0104] A number of embodiments have been described. Nevertheless, it will be understood that various modifications may be made. For example, blocks in the flowcharts may be skipped or performed out of order and still produce desirable results. Different translation methods may be used. Accordingly, other embodiments are within the scope of the following claims.

Claims

1. A computer implemented method of generating phrase-based joint probability model from a parallel corpus comprising a plurality of sentences in a source language and a corresponding plurality of sentences in a target language; the method comprising:

   a) determining from the parallel corpus high frequency n-grams $(\vec{e_i})$ in E, and $(\vec{f_i})$ in F, where E and F comprise sentences in the source and target languages respectively;
   b) obtaining an initial phrase-based joint probability distribution t, by:

      i) for each sentence pair (E,F) in the corpus, taking thee Cartesian product of n-grams $(\vec{e_i})$ in E, and $(\vec{f_i})$ in F;
      ii) for each n-gram pair (e_i, f_i) in the Cartesian product determining a t-count given by the expression:

$$\frac{\sum_{k=1}^{\min(l-a,m-b)} k!S(s(l-a,k)S(m-b,k))}{\sum_{k=1}^{\min(l,m)} k!S(s(l,k)S(m,k))}$$

wherein l and m are the lengths of the sentences E and F respectively, a and b are the lengths of the n-grams ($\vec{e}_i$) and ($\vec{f}_i$), and S is the Stirling Number of the second kind;
iii) summing over the t-counts and normalising; and

c) performing Expectation Maximum training for a plurality of iterations to generate a joint probability distribution t.

2. The method of claim 1, comprising repeating steps a) to c) using unigrams in place of n-grams.

3. The method of claim 1 or 2, comprising generating a conditional probability model from the joint probability model, wherein the conditional probability model can be used subsequently for decoding.

4. The method of claim 1 or 2, further comprising:

generating a phrase-to-phrase translation lexicon from the joint probability model and the parallel corpus.

5. The method of claim 4, wherein the phrase-to-phrase translation lexicon is generated, by:

i) stochastically generating a bag of concepts *C*;
ii) generating and discovering a single set of hidden concepts $c_i \in C$, whereby each concept generates a pair of phrases ($\vec{e}_i$, $\vec{f}_i$) according to the distribution t($\vec{e}_i$, $\vec{f}_i$) where each $\vec{e}_i$ and $\vec{f}_i$ contain at least one word; and
iii) ordering the phrases generated in each language so as to create two linear sequences of phrases.

6. The method of claims 4, wherein the phrase-to-phrase translation lexicon is generated by:

(1) stochastically generating a bag of concepts *C*;
(2) initialising E and F to empty sentences ε;
(3) randomly removing a concept $c_i \in C$ and generating a pair of phrases ($\vec{e}_i$, $\vec{f}_i$) according to the distribution t ($\vec{e}_i$, $\vec{f}_i$) where each $\vec{e}_i$ and $\vec{f}_i$ contain at least one word;
(4) appending the phrase $\vec{f}_i$ to the end of F;
(5) inserting phrase $\vec{e}_i$ at position *l* in E provided that no other phrase occupies any of the positions between *l* and $l + |\vec{e}_i|$,

where $|\vec{e}_i|$ gives the length of the phrase $\vec{e}_i$; and
repeating steps (3) to (5) until C is empty.

7. The method of claim 1, comprising generating a phrase-to-phrase translation lexicon from a parallel corpus using word-for-word alignments in the parallel corpus and a phrase-based model.

8. The method of claim 7, wherein said generating comprises:

performing a word-to-word alignment on both sides of the parallel corpus to produce a plurality of word alignments; and
collecting a plurality of aligned phrase pairs that are consistent with word alignments in said plurality of word alignments.

9. The method of claim 8, further comprising:

estimating a phrase translation probability distribution from the collected phrase pairs by relative frequencies.

10. The method of claim 9, further comprising:

parsing both sides of the word-aligned parallel corpus with a syntactic parser to generate syntactic parse trees; and
for each aligned phrase pair, checking if both phrases are subtrees in the syntactic parse trees.

11. The method of claim 9, further comprising:

identifying a collected aligned phrase pair having a plurality of alignments; and
calculating a lexical weight for each of said plurality of alignments.

12. The method of claim 7, wherein said generating comprises:

performing bidirectional word-to-word alignment operations on the parallel corpus to generate two sets of word alignments.

13. The method of claim 12, further comprising:

identifying alignment points at intersections between the two sets of word alignments.

14. The method of claim 12, further comprising:

identifying alignment points at the union between the two sets of word alignments.

15. The method of any of claims 1 to 6, further comprising: determining a translation for an input sentence in the first language using a greedy decoding operation.

16. The method of claim 15, further comprising determining the best output sentence in a second language for an input sentence in a first language by
segmenting the input sentence into a sequence of phrase ;
translating each of said phrases into a phrase in the second language; and
reordering the output phrases.

17. The method of claim 16, wherein said reordering comprises reordering the output phrases using a relative distortion probability distribution.

18. The method of any of claims 1 to 6, further comprising:

determining a translation for an input sentence in the first language using a beam search algorithm.

19. The method of claim 2 or 3, comprising:

(1) receiving an input string including a plurality of words in a first language;
(2) creating an initial hypothesis is a second language, wherein the initial hypothesis represent a partial translation of the input string in the second language containing zero or more words;
(3) selecting a sequence from said plurality of words in the input string:
(4) selecting a possible phrase translation in the second language using the joint or conditional probability model for said selected sequence;
(5) attaching the possible phrase translation to the current hypothesis to produce an updated hypothesis;
(6) marking the words in said selected sequence as translated;
(7) storing the hypothesis sequence in a stack;
(8) updating a probability cost of the updated hypothesis;
(9) repeating steps (3) to (8) based on a size of the stack to produce one or more possible translations for the input string; and
(10) selecting one of said possible translations in the stack having the highest probability.

20. The method of claim 19, wherein the each of the possible translations comprises a hypothesis leaving no corresponding untranslated words in the input string.

21. The method of claim 19, wherein said updating the probability cost comprises calculating a current cost for the updated hypothesis and estimating a future cost for the updated hypothesis.

22. The method of claim 21, further comprising:

discarding an updated output sequence if said updated hypothesis has a higher cost than n-best hypotheses in the stack, where n corresponds to a predetermined beam size.

23. The method of any preceding claim wherein the EM training is Viterbi based EM training.

Patentansprüche

1. Computerimplementiertes Verfahren zum Generieren eines gemeinsamen phrasenbasierten Wahrscheinlichkeitsmodells aus einem parallelen Korpus umfassend eine Mehrzahl von Sätzen in einer Quellsprache und eine entsprechende Mehrzahl von Sätzen in einer Zielsprache;
wobei das Verfahren Folgendes umfasst:

a) Bestimmen von hochfrequenten n-Grammen $(\vec{e}_i)$ in E und $(\vec{f}_j)$ in F aus dem parallelen Korpus, wobei E und F Sätze in der Quellsprache bzw. Zielsprache umfassen;
b) Erhalten einer anfänglichen gemeinsamen phrasenbasierten Wahrscheinlichkeitsverteilung t, indem

i) für jedes Satzpaar (E, F) in dem Korpus das Kartesische Produkt von n-Grammen $(\vec{e}_f)$ in E und $(\vec{f}_j)$ in F genommen wird;
ii) für jedes n-Gramm-Paar $(\vec{e}_i, \vec{f}_j)$ in dem Kartesischen Produkt ein t-Zählwert bestimmt wird, der gegeben ist durch den Ausdruck

$$\frac{\sum_{k=1}^{\min(l-a,m-b)} k!S(s(l-a,k)S(m-b,k))}{\sum_{k=1}^{\min(l,m)} k!S(s(l,k)S(m,k))}$$

wobei l und m die Längen der Sätze E bzw. F sind, a und b die Längen der n-Gramme $(\vec{e}_f)$ und $(\vec{f}_j)$ sind und S die Stirling-Zahl der zweiten Art ist;
iii) Summieren der t-Zählwerte und Normieren und

c) Ausführen eines Erwartungsmaximumstrainings für eine Mehrzahl von Iterationen zum Generieren einer gemeinsamen Wahrscheinlichkeitsverteilung t.

2. Verfahren nach Anspruch 1, umfassend das Wiederholen der Schritte a) bis c) unter Verwendung von Unigrammen anstelle von n-Grammen.

3. Verfahren nach Anspruch 1 oder 2, umfassend das Generieren eines bedingten Wahrscheinlichkeitsmodells aus dem gemeinsamen Wahrscheinlichkeitsmodell, wobei das bedingte Wahrscheinlichkeitsmodell später zum Decodieren verwendet werden kann.

4. Verfahren nach Anspruch 1 oder 2, weiterhin umfassend:

Generieren eines Phrase-zu-Phrase-Übersetzungslexikons aus dem gemeinsamen Wahrscheinlichkeitsmodell und dem parallelen Korpus.

5. Verfahren nach Anspruch 4, wobei das Phrase-zu-Phrase-Übersetzungslexikon generiert wird durch:

i) stochastisches Generieren eines Sacks von Konzepten C;
ii) Generieren und Entdecken einer einzelnen Menge von versteckten Konzepten $c_i \in C$, wodurch jedes Konzept ein Paar von Phrasen $(\vec{e}_i, \vec{f}_i)$ gemäß der Verteilung $(\vec{e}_i, \vec{f}_i)$ generiert, wobei jedes $\vec{e}_i$ und $\vec{f}_i$ mindestens ein Wort enthält; und
iii) Ordnen der in jeder Sprache generierten Phrasen, um zwei lineare Sequenzen von Phrasen zu erzeugen.

6. Verfahren nach Anspruch 4, wobei das Phrase-zu-Phrase-Übersetzungslexikon generiert wird durch:

(1) stochastisches Generieren eines Sacks von Konzepten C:
(2) Initialisieren von E und F zu leeren Sätzen ε;
(3) zufälliges Entfernen eines Konzepts $c_i \in C$ und Generieren eines Paars von Phrasen $(\vec{e}_i, \vec{f}_i)$ gemäß der Verteilung t $(\vec{e}_i, \vec{f}_i)$ wobei jedes $\vec{e}_i$ und mindestens ein Wort enthält;
(4) Anhängen der Phrase $\vec{f}_i$ an das Ende von F;
(5) Einsetzen der Phrase $\vec{e}_i$ an der Position l in E, vorausgesetzt keine andere Phrase besetzt eine der Positionen zwischen l und l + $|\vec{e}_i|$,

wobei $|\vec{e}_i|$ die Länge der Phrase $\vec{e}_i$ angibt; und
Wiederholen der Schritte (3) bis (5), bis C leer ist.

7. Verfahren nach Anspruch 1, umfassend das Generieren eines Phrase-zu-Phrase-Übersetzungslexikons aus einem parallelen Korpus unter Verwendung von Wort-für-WortAbgleichungen in dem parallelen Korpus und eines phrasenbasierten Modells.

8. Verfahren nach Anspruch 7, wobei das Generieren Folgendes umfasst:

Ausführen einer Wort-zu-Wort-Abgleichung auf beiden Seiten des parallelen Korpus, um eine Mehrzahl von Wortabgleichungen zu erzeugen; und
Sammeln einer Mehrzahl von abgeglichenen Phrasenpaaren, die mit Wortabgleichungen in der Mehrzahl von Wortabgleichungen übereinstimmen.

9. Verfahren nach Anspruch 8, weiterhin umfassend:

Schätzen einer Phrasenübersetzungswahrscheinlichkeitsverteilung anhand der gesammelten Phrasenpaare nach relativen Frequenzen.

10. Verfahren nach Anspruch 9, weiterhin umfassend:

Parsen beider Seiten des an Wortgrenzen abgeglichenen parallelen Korpus mit einem syntaktischen Parser zum Generieren syntaktischer Parse-Bäume und
Prüfen für jedes abgeglichene Phrasenpaar, ob beide Phrasen Teilbäume in den syntaktischen Parse-Bäumen sind.

11. Verfahren nach Anspruch 9, weiterhin umfassend:

Identifizieren eines gesammelten abgeglichenen Phrasenpaars mit einer Mehrzahl von Abgleichungen und Berechnen eines lexikalischen Gewichts für jede der Mehrzahl von Abgleichungen.

12. Verfahren nach Anspruch 7, wobei das Generieren Folgendes umfasst:

Ausführen von bidirektionalen Wort-zu-Wort-Abgleichungsoperationen an dem parallelen Korpus, um zwei Mengen von Wortabgleichungen zu generieren.

13. Verfahren nach Anspruch 12, weiterhin umfassend:

Identifizieren von Abgleichungspunkten an Schnittmengen zwischen den beiden Mengen von Wortabgleichungen.

14. Verfahren nach Anspruch 12, weiterhin umfassend:

Identifizieren von Abgleichungspunkten an der Vereinigungsmenge zwischen den beiden Mengen von Wortabgleichungen.

15. Verfahren nach einem der Ansprüche 1 bis 6, weiterhin umfassend:

Bestimmen einer Übersetzung für einen eingegebenen Satz in der ersten Sprache unter Verwendung einer Greedy-Decodieroperation.

16. Verfahren nach Anspruch 15, weiterhin umfassend:

Bestimmen des besten ausgegebenen Satzes in einer zweiten Sprache für einen eingegebenen Satz in einer ersten Sprache durch
Segmentieren des eingegebenen Satzes in eine Sequenz von Phrasen;
Übersetzen jeder der Phrasen in eine Phrase in der zweiten Sprache und
Neuordnen der ausgegebenen Phrasen.

17. Verfahren nach Anspruch 16, wobei das Neuordnen das Neuordnen der ausgegebenen Phrasen unter Verwendung einer relativen Verzerrungswahrscheinlichkeitsverteilung umfasst.

18. Verfahren nach einem der Ansprüche 1 bis 6, weiterhin umfassend:

Bestimmen einer Übersetzung für einen eingegebenen Satz in der ersten Sprache unter Verwendung eines Strahisuch-Algorithmus.

19. Verfahren nach Anspruch 1, 2 oder 3, umfassend:

(1) Empfangen einer eingegebenen Kette mit einer Mehrzahl von Wörtern in einer ersten Sprache;
(2) Erzeugen einer Anfangshypothese in einer zweiten Sprache, wobei die Anfangshypothese eine teilweise Übersetzung der eingegebenen Kette in der zweiten Sprache darstellt, die null oder mehr Wörter enthält;
(3) Wählen einer Sequenz aus der Mehrzahl von Wörtern in der eingegebenen Kette;
(4) Wählen einer möglichen Phrasenübersetzung in der zweiten Sprache unter Verwendung des gemeinsamen oder bedingten Wahrscheinlichkeitsmodells für die gewählte Sequenz;
(5) Anhängen der möglichen Phrasenübersetzung an die aktuelle Hypothese, um eine aktualisierte Hypothese zu erzeugen;
(6) Markieren der Wörter in der gewählten Sequenz als übersetzt;
(7) Speichern der Hypothesensequenz in einem Stapel;
(8) Aktualisieren von Wahrscheinlichkeitskosten der aktualisierten Hypothese;
(9) Wiederholen der Schritte (3) bis (8) auf der Basis einer Größe der Stapels, um eine oder mehrere mögliche Übersetzungen für die eingegebene Kette zu erzeugen; und
(10) Wählen einer der möglichen Übersetzungen in dem Stapel mit der höchsten Wahrscheinlichkeit.

20. Verfahren nach Anspruch 19, wobei jede der möglichen Übersetzungen eine Hypothese umfasst, die keine entsprechenden unübersetzten Wörter in der eingegebenen Kette zurücklässt.

21. Verfahren nach Anspruch 19, wobei das Aktualisieren der Wahrscheinlichkeitskosten das Berechnen von aktuellen Kosten für die aktualisierte Hypothese und das Schätzen von zukünftigen Kosten für die aktualisierte Hypothese umfasst.

22. Verfahren nach Anspruch 21, weiterhin umfassend:

Verwerfen einer aktualisierten ausgegebenen Sequenz, wenn die aktualisierte Hypothese höhere Kosten als n-beste Hypothesen in dem Stapel aufweist, wobei n einer vorbestimmten Strahlgröße entspricht.

23. Verfahren nach einem vorhergehenden Anspruch, wobei das EM-Training ein Viterbibasiertes EM-Training ist.

Revendications

1. Procédé mis en oeuvre par ordinateur pour générer un modèle de probabilité jointe basé sur des expressions venant d'un corpus parallèle constitué d'une pluralité de phrases dans une langue source et une pluralité correspondante de phrases dans une langue cible ;
le procédé comprenant les étapes suivantes :

a) déterminer dans le corpus parallèle des n-grammes haute fréquence ($\vec{e}_i$) dans E, et ($\vec{f}_i$) dans F, où E et F contiennent des phrases dans les langues source et cible respectivement ;
b) obtenir une distribution de probabilité jointe basée sur des expressions t en :

i) pour chaque paire de phrases (E, F) dans le corpus, prenant le produit cartésien des n-grammes ($\vec{c}_i$) dans E, et ($\vec{f}_i$) dans F ;
ii) pour chaque paire de n-grammes ($\vec{e}_i$, $\vec{f}_i$) dans le produit cartésien, déterminant un nombre t donné par l'expression :

$$\frac{\sum_{k=1}^{\min(l-a,m-b)} k!S(s(l-a,k)S(m-b,k))}{\sum_{k=1}^{\min(l,m)} k!S(s(l,k)S(m,k))}$$

dans lequel l et m sont les longueurs des phrases E et F respectivement, a et b sont les longueurs des n-grammes ($\vec{e}_i$) et ($\vec{f}_i$), et S est le nombre de Stirling du deuxième type ;
iii) totalisant les nombres t et en normalisant : et

c) réalisant un apprentissage Espérance-Maximisation (EM) pour une pluralité d'itérations afin de générer une distribution de probabilité jointe t.

2. Procédé selon la revendication 1, comprenant la répétition des étapes a) à c) en utilisant des unigrammes à la place de n-grammes.

3. Procédé selon la revendication 1 ou 2, comprenant la génération d'un modèle de probabilité conditionnelle à partir du modèle de probabilité jointe, dans lequel le modèle de probabilité conditionnelle peut être utilisé par la suite pour un décodage.

4. Procédé selon la revendication 1 ou 2, comprenant en outre :

la génération d'un lexique de traduction expression à expression à partir du modèle de probabilité jointe et du corpus parallèle.

5. Procédé selon la revendication 4, dans lequel le lexique de traduction expression à expression est généré en :

i) générant de manière stochastique un lot de concepts C ;
ii) générant et découvrant un seul jeu de concepts cachés $C_i \in C$, ce qui permet à chaque concept de générer une paire d'expressions ($\vec{c}_i$, $\vec{f}_i$) suivant la distribution t ($\vec{c}_i$,$\vec{f}_i$) où $\vec{c}_i$ et $\vec{f}_i$ contiennent chacun au moins un mot ; et
iii) ordonnançant les expressions générées dans chaque langue de manière à créer deux séquences linéaires d'expressions.

6. Procédé selon la revendication 4, dans lequel le lexique de traduction expression à expression est généré en :

(1) générant de manière stochastique un lot de concepts C ;
(2) initialisant E et F en phrases vides ε :
(3) supprimant aléatoirement un concept Ci ∈ C et générant une paire d'expressions ($\vec{e}_i$, $\vec{f}_i$) suivant la distribution t ($\vec{c}_i$, $\vec{f}_i$) où $\vec{e}_i$ et contiennent chacun au moins un mot ;
(4) annexant l'expression à l'extrémité de F ;
(5) insérant l'expression $\vec{e}_i$, à la position l dans E à condition qu'aucune autre expression n'occupe l'une quelconque des positions entre l et l + |$\vec{e}_i$| ,

où $|e_i|$ indique la longueur de l'expression $\vec{e}_i$, et en répétant les étapes (3) à (5) jusqu'à ce que C soit vide.

7. Procédé selon la revendication 1, comprenant la génération d'un lexique de traduction expression à expression à partir d'un corpus parallèle en utilisant des alignements mot à mot dans le corpus parallèle et un modèle basé sur des expressions.

8. Procédé selon la revendication 7, dans lequel ladite génération comprend :

la réalisation d'un alignement mot à mot des deux côtés du corpus parallèle pour produire une pluralité d'alignements de mots ; et
la collecte d'une pluralité de paires d'expressions alignées qui sont cohérentes avec les alignements de mots dans ladite pluralité d'alignements de mots.

9. Procédé selon la revendication 8, comprenant en outre :

l'estimation d'une distribution de probabilité de traduction d'expressions à partir des paires d'expressions collectées par fréquences relatives.

10. Procédé selon la revendication 9, comprenant en outre :

une analyse des deux côtés du corpus parallèle aligné suivant les mots avec un analyseur syntactique afin de générer des arbres d'analyse syntactique ; et
pour chaque paire d'expressions alignées, vérifier si les deux expressions sont des sous-arbres dans les arbres d'analyse syntactique.

11. Procédé selon la revendication 9, comprenant en outre :

l'identification d'une paire d'expressions alignées collectées ayant une pluralité d'alignements ; et
calculer un poids lexical pour chaque alignement dans ladite pluralité d'alignements.

12. Procédé selon la revendication 7, dans lequel ladite génération comprend :

la réalisation d'opérations d'alignement bidirectionnel mot à mot sur le corpus parallèle pour générer deux jeux d'alignements de mots.

13. Procédé selon la revendication 12, comprenant en outre :

l'identification de points d'alignement à des intersections entre les deux jeux d'alignements de mots.

14. Procédé selon la revendication 12, comprenant en outre :

l'identification de points d'alignement à l'union entre les deux jeux d'alignements de mots.

15. Procédé selon l'une quelconque des revendications 1 à 6, comprenant en outre : la détermination d'une traduction pour une phrase en entrée dans la première langue au moyen d'une opération de décodage glouton.

16. Procédé selon la revendication 15, comprenant en outre
la détermination de la meilleure phrase en sortie dans une deuxième langue pour une phrase en entrée dans une première langue en
segmentant la phrase en entrée en une séquence d'expressions ;
traduisant chacune desdites expressions en une expression dans la deuxième langue ; et
réordonnançant les expressions en sortie.

17. Procédé selon la revendication 16, dans lequel ledit réordonnancement comprend le réordonnancement des expressions en sortie au moyen d'une distribution de probabilité de distorsion relative.

18. Procédé selon l'une quelconque des revendications 1 à 6, comprenant en outre :

la détermination d'une traduction pour une phrase en entrée dans la première langue au moyen d'un algorithme de recherche en faisceau.

19. Procédé selon la revendication 1, 2 ou 3, comprenant :

(1) la réception d'une chaîne en entrée incluant une pluralité de mots dans une première langue;

(2) la création d'une hypothèse initiale dans une deuxième langue, dans laquelle l'hypothèse initiale représente une traduction partielle de la chaîne en entrée dans la deuxième langue contenant aucun ou plusieurs mot(s) ;
(3) la sélection d'une séquence dans ladite pluralité de mots dans la chaîne en entrée :
(4) la sélection d'une traduction d'expression possible dans la deuxième langue au moins du modèle de probabilité jointe ou conditionnelle pour ladite séquence sélectionnée ;
(5) l'association de la traduction d'expression possible à l'hypothèse actuelle pour produire une hypothèse mise à jour ;
(6) le marquage comme « traduits » des mots dans ladite séquence sélectionnée ;
(7) le stockage de la séquence de l'hypothèse en une pile ;
(8) la mise à jour d'un coût de probabilité de l'hypothèse mise à jour ;
(9) la répétition des étapes (3) à (8) sur la base d'une taille de la pile pour produire une ou plusieurs traduction (s) possible(s) pour la chaîne en entrée ; et
(10) la sélection de l'une des dites traductions possibles dans la pile ayant la probabilité la plus élevée.

20. Procédé selon la revendication 19, dans lequel chacune des traductions possibles comprend une hypothèse ne laissant aucun mot correspondant non traduit dans la chaîne en entrée.

21. Procédé selon la revendication 19, dans lequel la mise à jour du coût de probabilité comprend le calcul d'un coût actuel pour l'hypothèse mise à jour et l'estimation d'un coût futur pour l'hypothèse mise à jour.

22. Procédé selon la revendication 21, comprenant en outre :

l'élimination d'une séquence de sortie mise à jour si ladite hypothèse mise à jour a un coût plus élevé que les n meilleures hypothèses dans la pile, n correspondant à une taille de faisceau prédéterminée.

23. Procédé selon l'une quelconque des précédentes revendications, dans lequel l'apprentissage Espérance-Maximisation (EM) est un apprentissage EM basé sur l'algorithme de Viterbi.

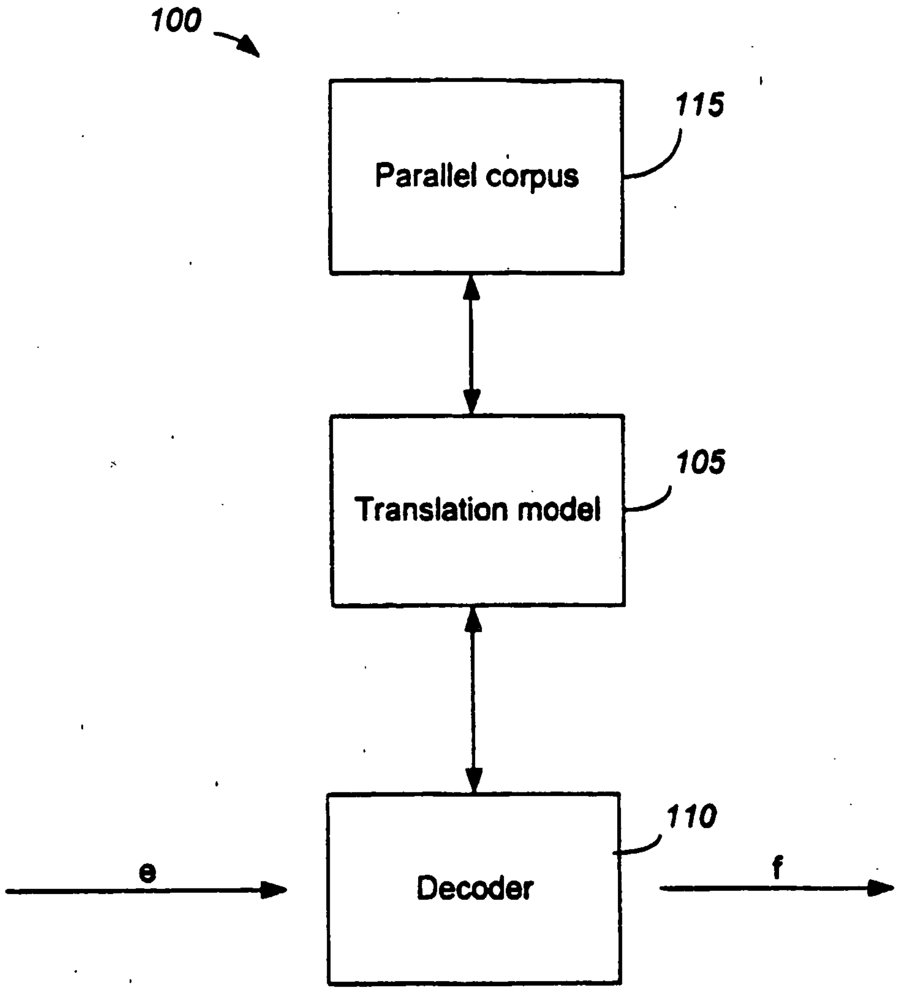
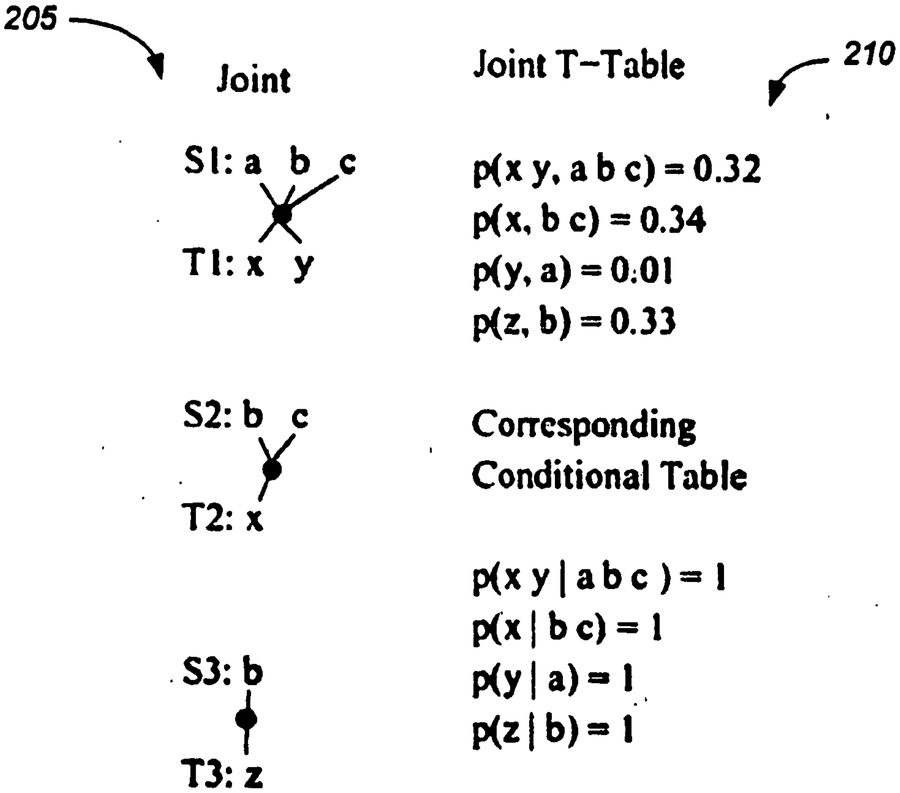

100
Parallel corpus
115
Translation model
105
Decoder
110
e
f

FIG. 1



Joint T-Table

p(x y, a b c) = 0.32
p(x, b c) = 0.34
p(y, a) = 0.01
p(z, b) = 0.33

Corresponding Conditional Table

p(x y | a b c) = 1
p(x | b c) = 1
p(y | a) = 1
p(z | b) = 1

FIG. 2

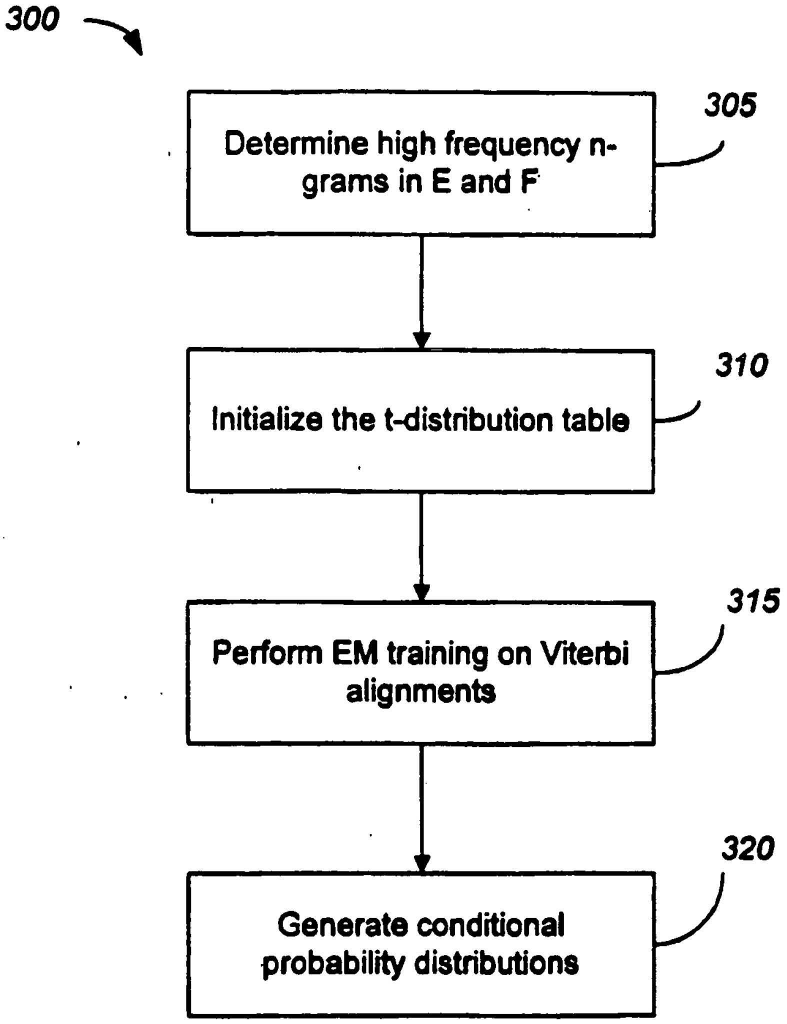

300
305
Determine high frequency n-grams in E and F
310
Initialize the t-distribution table
315
Perform EM training on Viterbi alignments
320
Generate conditional probability distributions

FIG. 3

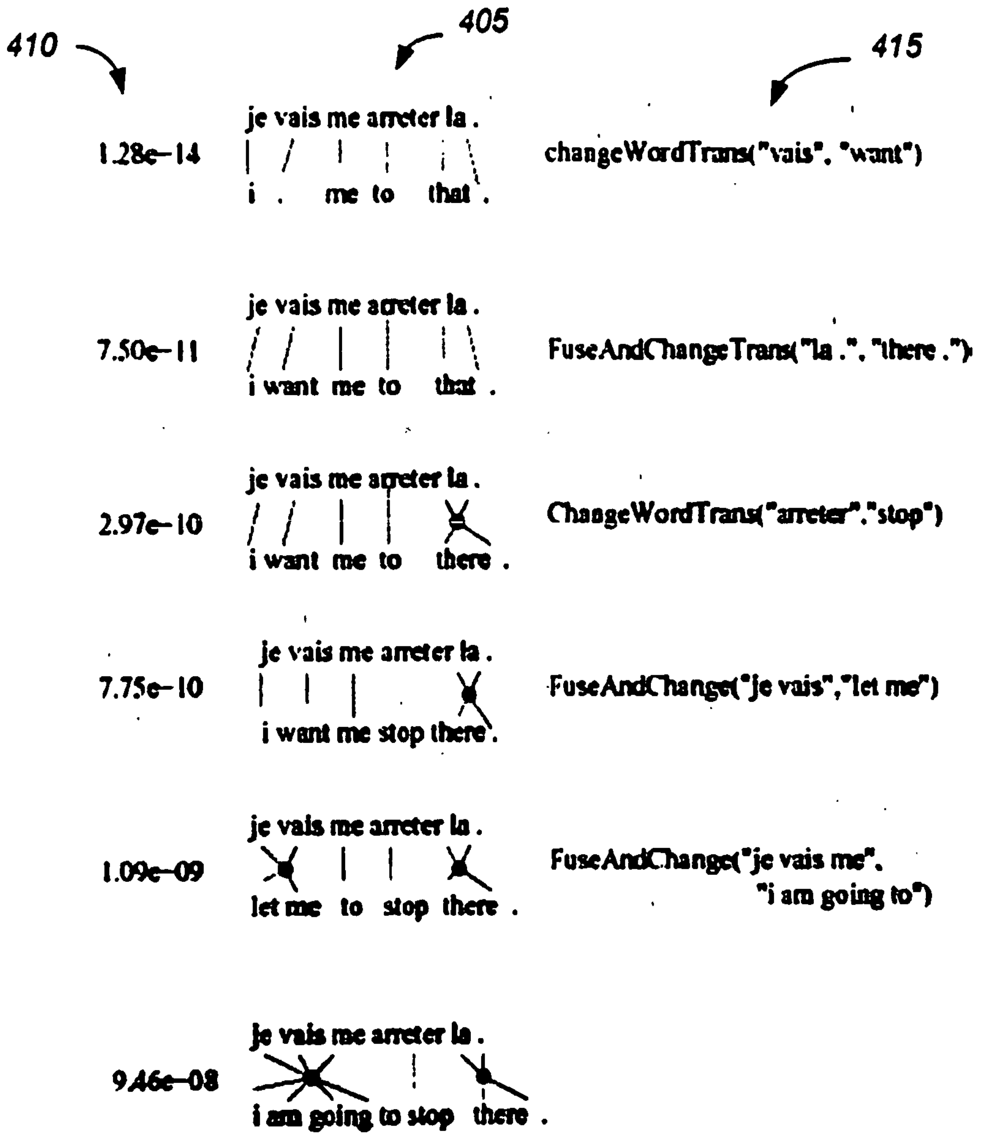

410
405
415
1.28e-14
je vais me arreter la .
i . me to that .
changeWordTrans("vais", "want")
7.50e-11
je vais me arreter la .
i want me to that .
FuseAndChangeTrans("la .", "there .")
2.97e-10
je vais me arreter la .
i want me to there .
ChangeWordTrans("arreter","stop")
7.75e-10
je vais me arreter la .
i want me stop there .
FuseAndChange("je vais","let me")
1.09e-09
je vais me arreter la .
let me to stop there .
FuseAndChange("je vais me", "i am going to")
9.46e-08
je vais me arreter la .
i am going to stop there .

FIG. 4

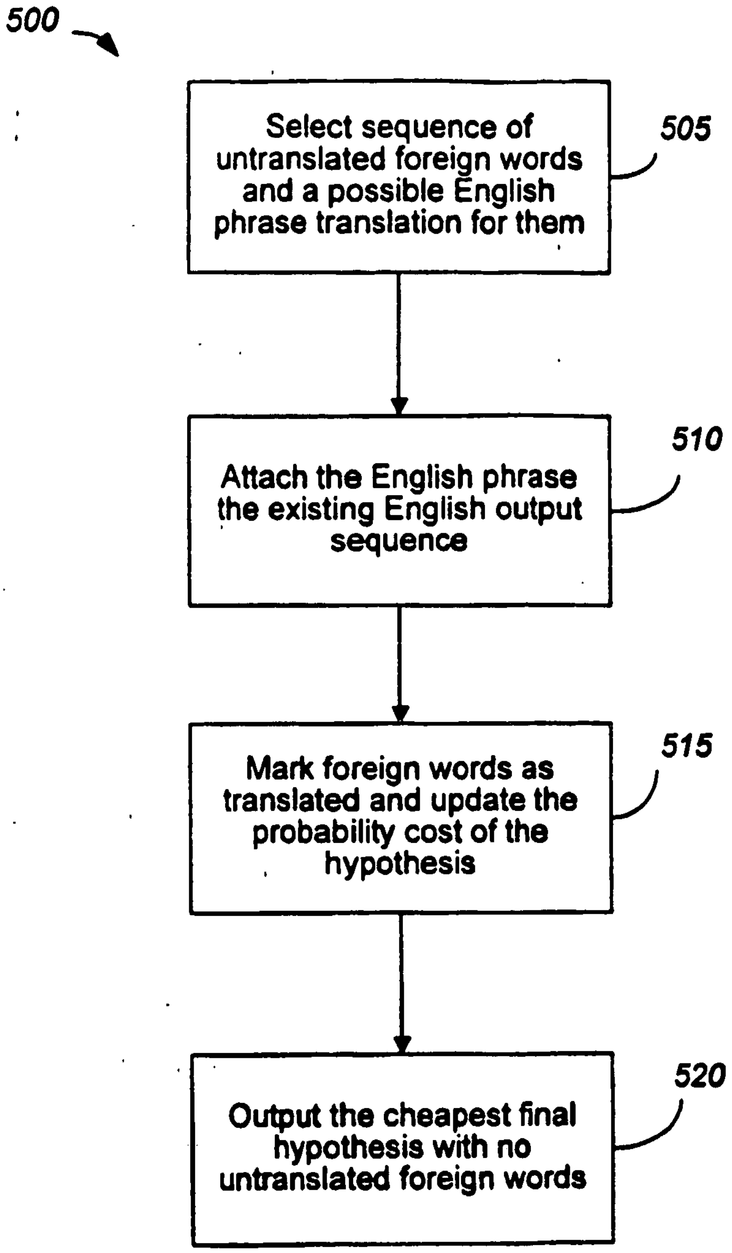

500
Select sequence of untranslated foreign words and a possible English phrase translation for them
505
Attach the English phrase the existing English output sequence
510
Mark foreign words as translated and update the probability cost of the hypothesis
515
Output the cheapest final hypothesis with no untranslated foreign words
520

FIG. 5

```
initialize hypothesisStack[0 .. nf];
create initial hypothesis hyp_init;
add to stack hypothesisStack[0];
for i=0 to nf-1:
  for each hyp in hypothesisStack[i]:
    for each new_hyp that can be derived from hyp:
      nf[new_hyp] = number of foreign words covered by new_hyp;
      add new_hyp to hypothesisStack[nf[new_hyp]];
      prune hypothesisStack[nf[new_hyp]];
find best hypothesis best_hyp in hypothesisStack[nf];
output best path that leads to best_hyp;
```

FIG. 6

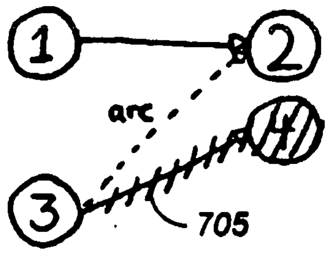


FIG. 7

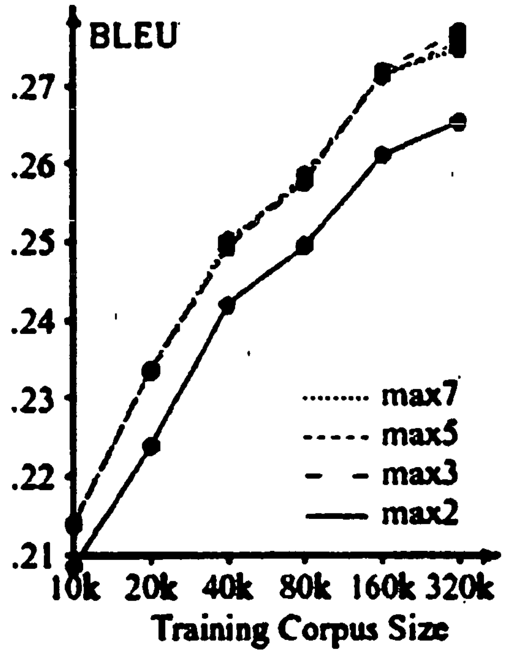

BLEU
.27
.26
.25
.24
.23
.22
.21
10k
20k
40k
80k
160k
320k
Training Corpus Size
max7
max5
max3
max2

FIG. 8

	f1	f2	f3
NULL	--	--	##
e1	##	--	--
e2	--	##	--
e3	--	##	--

$$\begin{aligned} p_w(\bar{f}|\bar{e}, a) &= p_w(f_1 f_2 f_3|e_1 e_2 e_3, a) \\ &= w(f_1|e_1) \\ &\quad \times \frac{1}{2}(w(f_2|e_2) + w(f_2|e_3)) \\ &\quad \times w(f_3|\mathrm{NULL}) \end{aligned}$$

FIG. 9

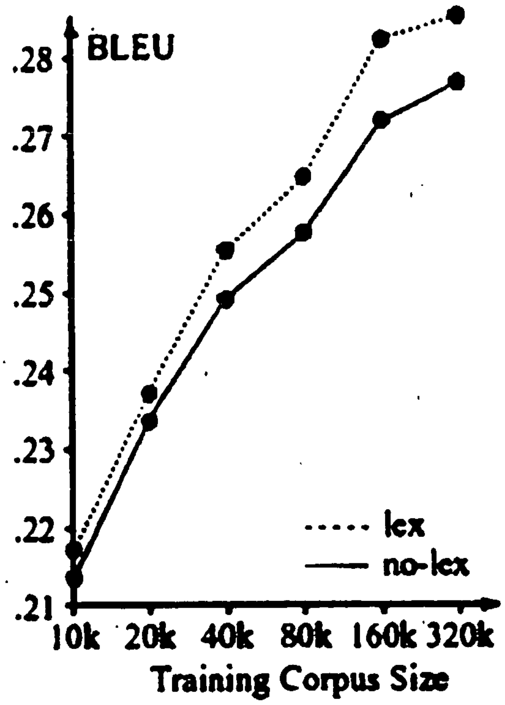

BLEU
.28
.27
.26
.25
.24
.23
.22
.21
10k
20k
40k
80k
160k
320k
Training Corpus Size
lex
no-lex

FIG. 10

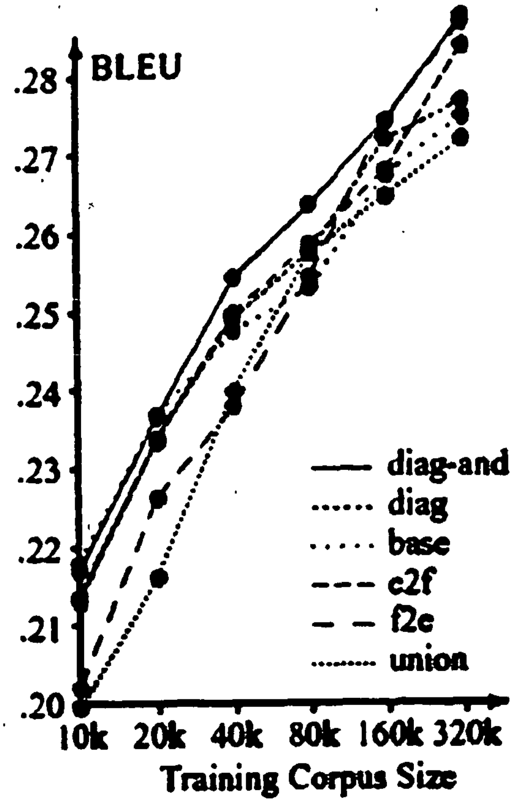

BLEU
.28
.27
.26
.25
.24
.23
.22
.21
.20
10k
20k
40k
80k
160k
320k
Training Corpus Size
diag-and
diag
base
e2f
f2e
union

FIG. 11

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Non-patent literature cited in the description

- **P Brown et al.** The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* **[0003]**
- **Dan Melamed.** Empirical Methods for Exploiting Parallel Texts. The MIT Press **[0003]**
- **Franz Josef Och et al.** Improved Alignment Models for Statistical Machine Translation. *Procedures of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora* **[0003]**