Available online at www.sciencedirect.com

SciVerse ScienceDirect

Computer Speech and Language 27 (2013) 438–454

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Unsupervised data processing for classifier-based speech translator☆

Emil Ettelaie*, Panayiotis G. Georgiou, Shrikanth S. Narayanan

Signal Analysis and Interpretation Laboratory, Ming Hsieh Department of Electrical Engineering, Viterbi School of Engineering, University of Southern California, 3710 S. McClintock Ave., RTH 320, Los Angeles, CA 90089, USA

Received 7 July 2010; received in revised form 17 February 2012; accepted 2 March 2012
Available online 12 March 2012

Abstract

Concept classification has been used as a translation method and has shown notable benefits within the suite of speech-to-speech translation applications. However, the main bottleneck in achieving an acceptable performance with such classifiers is the cumbersome task of annotating large amounts of training data. Any attempt to develop a method to assist in, or to completely automate, data annotation needs a distance measure to compare sentences based on the concept they convey. Here, we introduce a new method of sentence comparison that is motivated from the translation point of view. In this method the imperfect translations produced by a phrase-based statistical machine translation system are used to compare the concepts of the source sentences. Three clustering methods are adapted to support the concept-base distance. These methods are applied to prepare groups of paraphrases and use them as training sets in concept classification tasks. The statistical machine translation is also used to enhance the training data for the classifier which is crucial when such data are sparse. Experiments show the effectiveness of the proposed methods.

Keywords: Speech to speech translation; Spoken language understanding; Concept classification

1. Introduction

Phrase-based Statistical Machine Translation (SMT) methods (Ney et al., 2000; Och and Ney, 2004) are well established in the design of speech-to-speech (S2S) translation systems as the main translation technique (Narayanan et al., 2003; Hsiao et al., 2006). They are designed to operate based on local mapping of variable word sequences efficiently. Due to their flexibility these methods provide a good coverage of the dialog domain. The fluency of the translation, however, is not guaranteed. This is exacerbated by disfluency in the spoken input and potential recognition errors which creates a "noisy" utterance at the input of the translation unit. The lexical and syntax errors in the input of an SMT system often cause a severe degradation in the translation quality. All these ultimately affect the quality of the synthesized speech output in the target language, and the effectiveness of the concept transfer from the source to the target language. Therefore, it is quite common to use other means of translation in parallel to the SMT methods (Gao et al., 2006; Stallard et al., 2006).

The main goal in interactive S2S applications is to facilitate the accurate exchange of the concepts between the interlocutors rather than producing a word by word (literal) translation of the source. Existing SMT systems are typically

☆ This paper has been recommended for acceptance by Roger K. Moore.

* Corresponding author. Tel.: +1 2137402356.
E-mail address: ettelaie@usc.edu (E. Ettelaie).

0885-2308/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.csl.2012.03.001

optimized for faithful translation rather than this form of interpretation. A well defined structured dialog domain, e.g. doctor–patient dialog, can be partly covered by a number of concept classes. If a fluent representative sentence in target language is assigned to each concept class, the translation task simplifies to classifying the source utterances based on the concepts they convey. This well serves the purpose of exchanging concepts as long as the input utterance falls within the coverage of the classifier.

The process of concept classification can be viewed as a quantization of a continuous "semantic" sub-space. If the input utterance falls in the coverage domain of the quantizer and the selected concept matches it well, the quantization error is small and the classifier is adequate. Obviously it is not feasible to accurately cover the entire dialog domain with categorical concept classes. The reason is the large number of quantization levels that would be necessary due to the infinite number of concepts that could be exchanged in a dialog. A very large number of concept classes will also decrease the classifier accuracy. Therefore the classifier should be accompanied by a translation system with a much wider range such as an SMT engine. A rejection mechanism can help identify the cases that the input utterance falls outside the classifier coverage (Ettelaie et al., 2006).

In spite of the limited coverage, since the concept representatives are pre-selected well-formed sentences, the correct classification assures high quality of the output. Furthermore, the "hard decision" nature of the classification process makes the classifier engine more robust to input errors compared to SMT methods. Therefore, the classifier-based translator is an attractive option for S2S applications designed for structured dialog interactions with high levels of predictability. In addition, and even if it is used along side an SMT engine, it can provide the users with both an accurate feedback and different translation options to choose from. The latter feature, specially, is useful for applications like doctor–patient dialog. Moreover, the classification task, is much less computationally demanding compared to the statistical methods, and therefore, is more suitable for applications with small tolerable latency, or when the system is implemented on a small device.

The concept classification method has been successfully integrated in S2S translators as an alternative way of translation (Narayanan et al., 2003; Ehsani et al., 2006). It has been also used in a range of other applications such as systems with virtual interactive characters or machine spoken dialog systems to implement speech understanding (Leuski et al., 2006; Traum et al., 2007). Building a concept classifier starts with identifying the desired concepts and representing them with canonical utterances that express these concepts. Here, instead of an abstract definition of concepts, we have chosen a representation through a set of utterances that convey them. In practice this loose representation, eliminates an abstraction level in both data processing and classification task.

A good set of concepts should consist of the ones that are more frequent in a typical interaction in the domain. For instance in a doctor–patient dialog, the utterance "Where does it hurt?" is quite common and therefore its concept is a good choice. Phrase books, websites, and experts' judgment are some of the resources that can be used for concept selection. Other frequently used concepts include those that correspond to basic communicative and social aspects of the interaction such as greeting, acknowledgment and confirmation, encoded typically dialog acts.

After forming the concept space, utterances that convey the concept of each class must be gathered. Hence, this training corpus would consist of a group of paraphrases for each class. This form of data are often very difficult to collect as the number of classes grow.

Currently available data are collected from variety of sources in different domains. The realistic nature of these corpora is a great incentive to seek automatic ways of both selecting the concept classes and automatic clustering of the training data into these classes. In other words, we need a process through which the collected utterances are clustered based on the concept that they carry. To achieve that, a measure of similarity (or distance metric) between utterances is essential. We introduce a method of similarity assessment that is motivated by the translation task. In this method an SMT engine was deployed as an information source to enhance the method of comparing utterances (Ettelaie et al., 2008). With the existence of such a distance metric for comparing the data utterances, the problem of unsupervised training of the classifier would simply reduce to a clustering problem. The main focus of this work is twofold: (1) identifying a cross-sentence distance metric that will correlate well with the concept closeness of the two sentences in question, and (2) identifying and employing clustering techniques that rely on relative rather than global distance metrics.

Since the classifier range is limited, high accuracy within that range is quite crucial for its effectiveness. Some techniques have already been proposed to improve the classification rates. For example in Ettelaie et al. (2006) the accuracy has been improved by introducing a dialog model. Using hierarchical structures can also help achieve a better accuracy (Ettelaie et al., 2010). When the available data resources are limited (Narayanan et al., 2004), e.g. few

sentences per class, it is vital to use robust models and extra processing to overcome the low classification accuracy caused by sparsity of the training data. For instance a background model can be used to improve the discrimination ability of a given concept class model (Ettelaie et al., 2006). Here, we show that the data sparsity issue can be also handled by using an SMT engine as an information source (Ettelaie et al., 2008). Also, the effect of the background model on classification accuracy is investigated further.

The following section reviews the translation technique by means of concept classification, along with its training procedure. A method for coping with data sparsity based on using SMT engine is introduced in Section 3. In Section 4 the proposed method for unsupervised training is explained in detail. Section 5 covers the evaluation of the training procedure. Both intermediate and end-to-end evaluation measures are discussed. Section 6 consists of the experiment details and the associated data used in this work. The proposed method of unsupervised training and the application of the *k*-means clustering algorithm were both investigated. The results are compared and discussed in Section 7 which is followed by conclusion in Section 8.

2. Concept classifier

The concept classifier based on the maximum likelihood criterion can be implemented as a language model (LM) scoring process. For each class a language model is built using data expressing the class concept. The classifier scores the input utterance using the class LM's and selects the class with highest score. In another word if $\mathcal{C}$ is the set of concept classes and $\mathbf{e}$ is the input utterance, the classification process is,

$$\hat{C} = arg\max_{C \in \mathcal{C}} \{P(\mathbf{e} \mid C)\} \tag{1}$$

where $P(\mathbf{e} \mid C)$ is approximated by the score of $\mathbf{e}$ from the LM of class C. The translation job is concluded by playing out a previously stored prompt that expresses the concept of class $\hat{C}$ in the target language.

It is clear that for a class with limited number of training utterances, the associated LM will have a poor coverage. Even with a large training set for each class, the class language models will have a large number of out of vocabulary words due to the fact that classes usually have a limited lexicon. In practice such a model fails to produce a usable LM score and leads to a poor classification accuracy. Interpolating the LM with a background language model results in a smoother model (Stolcke, 2002) and increases the overall accuracy of the classifier. The background model should be built from a larger corpus that fairly covers the domain vocabulary. The interpolation level can be optimized for the best performance based on held-out set.

While the training of an SMT is mainly done through the use of bilingual parallel data, to train a concept classifier, sets of sentences with same (or very similar) concept are required. These are often generated by deciding a set of canonical utterances and then for each one of them manually generating a large number of paraphrases with significantly similar concepts (Narayanan et al., 2004). This procedure can be extremely time consuming.

The LM-based method of Eq. (1) was used in the Transonics system (Belvin et al., 2005) which was developed as an English/Farsi speech translator in the doctor–patient interaction domain. For that system, the canonical concepts were manually selected using medical phrase books, websites, and by human judgment (Narayanan et al., 2004). Then paraphrases were collected from human subjects in different ways at *the Information Sciences Institute* of the *University of Southern California*. A website was design to collect on-line data by showing a sentences to the users and asking them to provide as many paraphrases as they could. A similar tool was developed to collect speech data by letting the users to say and record paraphrases. Also, an on-line game was designed in which the players were scored based on the number and quality of paraphrases they entered (Chklovski, 2005). Paraphrasing sessions were also held in which a small group of volunteers provided data for some of the concepts. The lessons learned can be summarized as follows:

1. Manually selecting the concepts can lead to a poor coverage of the dialog domain.
2. Since the selected concepts are not driven from real data, some of them might be uncommon in real dialogs.
3. Overlapping concepts are very difficult to avoid.
4. For moderate number of classes, training data are difficult to collect from human resources, and the method is not practical for large number of classes.
5. The paraphrases provided by human subjects are not always common sentences in the dialog domain.

However, the demonstrated suitability of classifier-based translation engines for the Transonics system is a strong incentive to seek new approaches to overcome some of the above obstacles. For instance, we observed that the participants in system evaluations usually selected the classifier output when the dialog context was within the classifier domain (medical).

3. Handling sparsity by statistical machine translation

The main idea of this section is to address how to overcome the data sparsity by automatically generating multiple utterances—even if of lower quality—from a single original one. These generated utterances are expected to have a level of grammatical and lexical errors and hence, we refer to them as "noisy" utterances (see Section 4.1).

One approach is to use an SMT to generate n-best lists of translation candidates for the original utterances (Ettelaie et al., 2008). Such lists are ranked based on a combination of scores from different models (Ney et al., 2000). The hypothesis here is that for an SMT trained on a large corpus, the quality of the candidates would not degrade rapidly as one moves down the n-best list. Therefore the majority of the candidates are expected to have acceptable quality as long as the length of the list is chosen appropriately. This process would result in more data, available for training, at the cost of using noisier data.

Although the source language of the SMT must be the same as the classifier's, its target language can be selected independently. It is clear that a language with large available resources (in the form of parallel corpora with the source language) must be selected. For simplicity this language is called the "intermediate language" here.

A classifier in the intermediate language can be built by first generating an n-best list for every source utterance in the classifier's training corpus. Then the n-best lists associated with each class are combined to form a new training set. The class LM's are now built from these training sets rather than the original sets of the source utterances.

To classify a source utterance $\mathbf{e}$, first the SMT is deployed to generate an n-best list (in the intermediate language) from it. The list will consist of candidates $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n$. The classification process can be reformulated as,

$$\hat{C} = \underset{C \in \mathcal{C}}{arg\max} \left\{ \prod_{i=1}^{n} P(\mathbf{f}_i \mid C) \right\} \tag{2}$$

Here, $P(\mathbf{f}_i \mid C)$ is approximated by the score of the ith candidate $\mathbf{f}_i$ from the LM of class C in the intermediate language. The scores are considered in the probability domain.

The new class LM's can also be smoothed by interpolation with a background model in the intermediate language.

4. Unsupervised data processing

The available data corpora for S2S applications cannot be directly used for the purpose of classifier training without additional processing. These corpora are usually collected from example conversations. At best, data can be transcribed to train system components such as monolingual language models for speech recognizer. Also, large portions of the collected data are often translated for SMT training (bilingual). The goal is to identify the common concepts in these data sets, and for each concept, create a cluster containing all the utterances that convey that concept. Human input could be employed at the final step to represent the concepts in a canonical form in the target language. Obvious requirements for implementing this procedure are first specifying a distance metric among phrases and second, choosing a clustering method.

4.1. Utterance level distance

Since the goal is translation, it is intuitive to think that if two sentences have similar translation in another language, they convey the same concept. Therefore the similarity of two sentences can be judged by the similarity of their translations. This motivates the definition of a distance metric that is based on comparing the translations. In fact, two utterances from the training corpus are put together in the same concept class because they have the same translation.

However, two sentences with the same concept might have translations that share no common word. Therefore such comparison must be based on multiple translations of each original sentence. Otherwise it would not be discriminative enough for any practical application. Using multiple translations can partially address problems like lexical mismatch.

If a group of translations is available for every data utterance, defining a distance metric for them can be seen as a document comparison problem.

Various methods of document comparison have been introduced and deployed in a wide range of applications. The essence of these methods is a measure that indicates the similarity of the documents. For instance, in text clustering, documents are represented by points in a vector space. For each document, a vector is generated in a fashion where words represent the dimensions and number of occurrences, the scales (Dhillon and Modha, 2001). Then, distance measures, e.g., Euclidean distance, can be used as a similarity metric. The vectors, however, will contain no word ordering information.

In practice the sparsity of such vectors would make the comparison inaccurate. For example the vocabulary of a medium sized domain could contain more than 6000 words (Narayanan et al., 2006), but a sentence, represented by an M-dimensional ($M > 6000$) vector $\mathbf{s}$, will contain at most 10–20 words. As a result, direct comparison of two such vectors would unlikely provide any meaningful distance measurement.

The proposed solution here, can be viewed as "fuzzifying" the vector representations of sentences by adding up a large number of noisy measurements to make a new representation vector. If $\mathbf{s}_i$ is the vector representing a valid utterance, a corrupted version of that utterance will be represented by a vector $\mathbf{x}_i$ that can be modeled as a noisy measurement ($\mathbf{x}_i = \mathbf{s}_i + \mathbf{v}_i$). The document consist of corrupted versions of n utterances is represented by vector $\mathbf{x}_1 + \ldots + \mathbf{x}_n$. The goal is to reduce the sparsity of the measurement while attempting to keep the noise – at the concept level – to a minimum.

One way to create such noisy measurements $\mathbf{x}_i$ is by employing an SMT engine in a language pair with plentiful available resources, i.e., in the level of million words. In that case $\mathbf{x}_i$ would be a representation for a sentence in the target or intermediate language.

Again the assumption here is that for an SMT built on a vast amounts of data, there would be little quality degradation as one moves further down the n-best list, if the length of the list is chosen appropriately. Therefore $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n$ will contain acceptably low levels of noise.

4.2. Distance measure

The classifier presented by Eq. (1) is aimed at comparing utterances based on class LM's. The documents can be also compared on an LM base, mainly to incorporate some level of word ordering information in the assessment of the distance metric. By building an LM for each n-best list, each utterance in the data would have an associated LM in the target (or intermediate) language. Since these models are approximations of probability density functions, they can be compared using information theoretic measures like relative entropy (*Kullback–Leibler Divergence* – KLD). Although relative entropy is not commutative and therefore could not be used as a metric, it can be modified in the following way to serve the purpose.

$$\mathrm{KLD}_{\mathrm{sym}}(P, Q) = \frac{1}{2} D(P \| Q) + \frac{1}{2} D(Q \| P) \tag{3}$$

Jensen–Shannon Divergence (JSD) (Lin, 1991) is another symmetric and smoother derivation of the relative entropy. It is defined as,

$$JSD(P, Q) = \frac{1}{2} D(P \| M) + \frac{1}{2} D(Q \| M) \tag{4}$$

where $M = (1/2)(P + Q)$. A recursive algorithm has been presented in Sethy et al. (2004) to efficiently calculate the relative entropy (and hence KLD and JSD) between language models.

4.3. Clustering algorithms

Selecting concepts and forming classes of data for classifier training is in fact a clustering problem. While each cluster is assumed to represent a concept class, the utterances forming these clusters can be used to train the classifier.

Hierarchical techniques (e.g., agglomerative methods) are not very popular in text clustering applications. This is mainly due to their inherent chaining effect which lets a very "dissimilar" item appear in a cluster just because of its closeness to another member. Experiments with the agglomerative algorithm in the early stages of this work did not show any promising results either.

The partitional algorithms that are common in document clustering mostly rely on the presentation of the items in a coordinate system. For instance *k*-means algorithm and all its variations are based on centroid computations that are only meaningful when the items are presented in a vector space. This is despite the variety of ways that the vectors might be defined and the variations in processing details such as normalization and inverse document frequency (idf) adjustment.

When the items are compared based on the language models' mismatch (Eqs. (3) and (4)), the clustering algorithm must take the distance among the items as the sole form of information. With no definition of item coordinates, centroid computation is not possible. Also, since the information theoretic measures do not satisfy triangular inequality, the clustering must not rely on such property.

We selected the *Exchange Method* and the *Affinity Propagation* clustering algorithms to use with the measures of Eqs. (3) and (4). Both methods only use the relative distances and were derived with no assumptions regarding the triangular inequality.

However, here, the goal is to use language models and an information theoretic measure as a similarity metric to compare them. In that case, the only algorithms that could be applied are the ones that rely on the distances among the items as the sole form of information, as the items' coordinates are not defined and centroid computation is not possible.

4.3.1. Exchange Method

The Exchange Method introduced in Späth (1985) reformulates the clustering problem as an optimization task. This algorithm has also been used for summarization (Hatzivassiloglou et al., 2001) and clustering semantically-related adjectives (McKeown and Hatzivassiloglou, 1993).

If $C_1, C_2, \ldots, C_K$ are the clusters, the goal is to minimize the cost function,

$$Phi(\mathcal{C}) = \sum_{i=1}^{K} \frac{1}{|C_i|} \sum_{\substack{x,\, y \in C_i \\ x \neq y}} d(x, y) \tag{5}$$

$$\mathcal{C} \triangleq \{C_1, C_2, \ldots, C_K\} \tag{6}$$

Here, $d(x, y)$ is the distance between items x and y, and $|\cdot|$ is the cluster cardinality. The algorithm is as follows:

1. Randomly assign one item to each cluster (to make sure no cluster is initialized as an empty one).
2. Assign the remaining items randomly to the clusters.
3. Select the first item.
4. If the selected item is the only member of its cluster select the next item.
5. Move the selected item from its cluster to the rest of them, and check the change in the cost of Eq. (5). If the movement lowers the cost keep the item in the new cluster. Total cost recalculation is not necessary after each movement. If item t belongs originally to cluster C_k (obviously $|C_k| > 1$), its movement to cluster C_i will decrease the total cost if and only if,

$$\frac{1}{|C_k|(|C_k|-1)} \left[\sum_{\substack{x,\, y \in C_k \\ x \neq y}} d(x, y) - |C_k| \sum_{\substack{x \in C_k \\ x \neq t}} d(x, t) \right] < \frac{1}{|C_i|(|C_i|+1)} \left[\sum_{\substack{x,\, y \in C_i \\ x \neq y}} d(x, y) - |C_i| \sum_{x \in C_i} d(x, t) \right]$$

   The above condition must be checked for every cluster $C_i \in \mathcal{C}, i \neq k$.
6. Repeat steps 4–5 for the next items until all of them are checked.
7. Start over from step 3 until no cost change is observed.

To avoid local minima, the above algorithm should be run several times with different random initialization in step 1.

4.3.2. Affinity Propagation

Affinity Propagation (Frey and Dueck, 2007) was introduced based on the max-sum algorithm in factor graphs (Kschischang et al., 2001). The clusters form gradually through an iterative message passing procedure.

Similar to k-centers algorithm, every cluster is represented by one of the data items called an "exemplar". Every item is initially a potential exemplar. As the algorithm proceeds, some of the items emerge as stronger exemplars and every other item shows tendency to be represented by one of these exemplars.

The input to the algorithm is a table of pair-wise distances between the data items and also a "preference" number that indicates each item's potential to become an exemplar. Every data item sends messages, called "responsibility", to the other items indicating how well it could be represented by each one of them. As a potential exemplar, each data item shows its adequacy to represent another item by sending it a message called "availability". In every iteration these two types of messages are exchanged among all the items.

If $\mathcal{D}$ is the set of all items, for two items $x, y \in \mathcal{D}$ with distance $d(x, y)$ between them, we show the responsibility and availability by $r(x, y)$ and $a(x, y)$, respectively, when y is considered the potential exemplar. The preferences of x is shown by $-d(x, x)$ (not a distance). The algorithm can be described as follows.

1. $\forall x, y \in \mathcal{D}$ initialize $a(x, y) = 0$
2. $\forall x, y \in \mathcal{D}$ update the responsibilities as,

$$r(x, y) \leftarrow \lambda \cdot r(x, y) + (1 - \lambda) \cdot \left[-d(x, y) - \max_{\substack{t \in \mathcal{D} \\ t \neq y}} \{a(x, t) - d(x, t)\} \right]$$

3. $\forall x, y \in \mathcal{D}$ update the availabilities as,
   if $x \neq y$

$$a(x, y) \leftarrow \lambda \cdot a(x, y) + (1 - \lambda) \cdot \min \left\{ 0, r(y, y) + \sum_{\substack{t \in \mathcal{D} \\ t \neq x, y}} \max\{0, r(t, y)\} \right\}$$

   else

$$a(x, x) \leftarrow \lambda \cdot a(x, x) + (1 - \lambda) \cdot \sum_{\substack{t \in \mathcal{D} \\ t \neq x}} \max\{0, r(t, x)\}$$

4. $\forall x \in \mathcal{D}$ update the exemplars as,

$$e_x \leftarrow \max_{t \in \mathcal{D}} \{a(x, t) + r(x, t)\}$$

5. While the stop criterion is not met, go to 2
6. $\forall x \in \mathcal{D} : e_x = x$, select the corresponding cluster as,

$$C^{(x)} = \{t \in \mathcal{D} : e_t = x\}$$

The damping factor λ was included in the update rules to guarantee the convergence of the algorithm (Frey and Dueck, 2007). An item x is an exemplar if $e_x = x$. The stop criterion can be selected as a fix number of iterations, or when the messages change below a threshold in two consequent iterations, or when the exemplars remain unchanged after a certain number of iterations.

The algorithm does not take the number of clusters as a prefixed parameter and determines it automatically. However, this number is mainly affected by the choice of the preferences. Higher preferences means more chance for the items to emerge as exemplars and therefore a higher number of clusters.

Since the algorithm does not need a random initialization, a one-time run suffices to get the optimum results and therefore significantly reduces the overall clustering time. Beside the fact that the triangular inequality is not necessary for the Affinity Propagation method, it is not even restricted to the commutative distance measures and therefore accepts measures like relative entropy with no modification.

4.4. Concept representatives

Each data cluster is associated with a concept that needs to be represented by a canonical utterance. A human supervisor can generate this representative or manually draw one from the cluster members. When the number of clusters is too large, a method similar to the one used in Ye and Young (2006) can be applied to select the representatives automatically, i.e.,

$$\forall C \in \mathcal{C} : r_C = \underset{x \in C}{\arg\min} \sum_{\substack{t \in C \\ t \neq x}} d(x, t) \tag{7}$$

In each cluster C, an utterance r_C is selected that has the least accumulative distance with the other members of that cluster. The two methods can be combined so that a few automatically selected utterances will be used by a human supervisor for identifying the concept representative.

4.5. Training

What was described in the above sub-sections, can be put together as an automatic method for concept selection and training data preparation for a concept classifier. The steps are illustrated in Fig. 1 and are implemented as follows,

1. *Domain definition*: utterances from the source language data are selected.
2. *Sparseness reduction*: an SMT system is used to translate these utterances to the target or any other language.
3. *Statistical representation*: for each source utterance, a language model is built from the associated n-best list provided by the SMT system.
4. *Distance metrics*: A table of JSD or KLD measures are built for all the possible language model pairs.
5. *Clustering*: using the above distance information, Exchange Method or Affinity Propagation is applied to cluster the original utterances.
6. *Representative selection*: for each cluster, a representative is chosen. The representative might be translated (manually or by an SMT) or pulled out from the target language part of data, in case of parallel corpus. If the classifier selects a certain class, the translated representative of that class would be the output of the overall system.

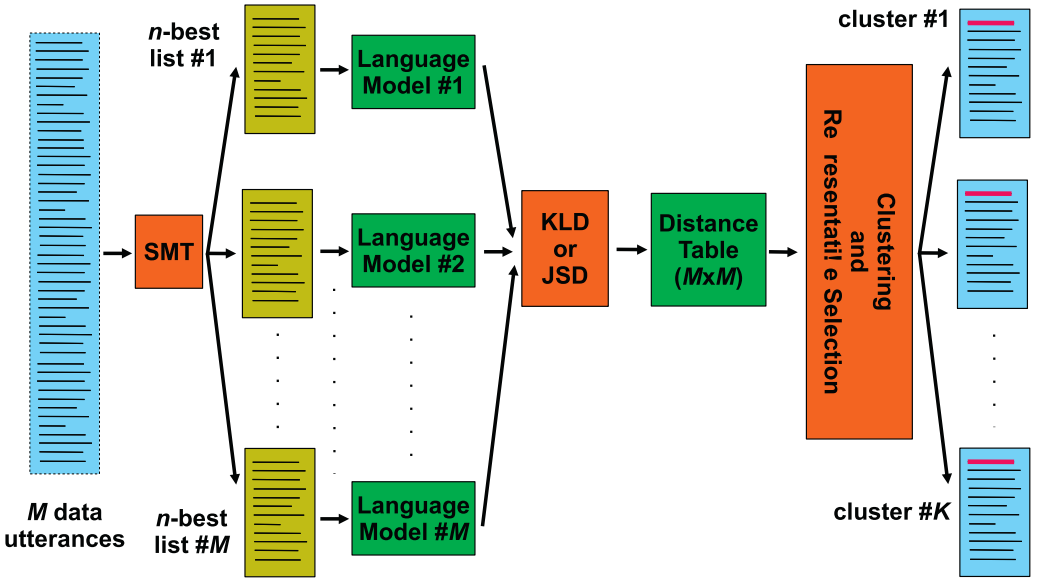

Fig. 1. Overview of the proposed data preparation procedure.

The complex data sentences can be broken into smaller segments through a preprocessing stage. This will reduce, but not eliminate, the number of utterances with multiple concepts. Presence of such utterances in large proportions may lead to clusters with ambiguous concepts.

5. Evaluation

The classifier accuracy is the main evaluation measure, however, measuring the clustering quality, as an intermediate level assessment is beneficial for developing the training method. Despite its popularity in the MT world, the BLEU score (Papineni et al., 2012) would not be a useful measure for concept classification task. BLEU score is based on the lexical matching of the hypothetical translation and its reference. However, two lexically disjoint utterances, could express the same concept. For instance, "have a seat" and "please sit down" match in concept while having no common word.

5.1. Clustering evaluation

Two methods for evaluating the quality of the clustering task have been used in this work. The first one is introduced in Wagstaff and Cardie (2000) and is based on computing the percentage of binary decisions that are common between the clustered and reference data. Every possible pair of data items, gives a correct/wrong output depending on the agreement of the reference with the hypothesis regarding to their same-cluster status. A percentage of the decisions that are in agreement with the reference data can be used as an indicator of the quality of the clustering task. In other words the agreement can be presented as,

$$Agr(\mathcal{C}, \mathcal{R}) = \frac{\text{agreement}(\mathcal{C}, \mathcal{R})}{\binom{N}{2}} \tag{8}$$

where agreement($\mathcal{C}$, $\mathcal{R}$) is the number of common binary decisions between cluster set $\mathcal{C}$ and reference $\mathcal{R}$, and N is the number of data items. The denominator is the number of possible pairs and therefore the same number of decisions are to be made.

Note that this measure considers placing two items from different classes in two different clusters, a correct decision, and hence is highly biased toward the correct measurements. For example, with 100 classes, a decision about two given items from different classes, has 99% chance of being right. Therefore, this measure saturates as the number of items

Table 1
The data sets used for clustering and classification.

Data set	Transonics	BBN
Language	English	English
Domain	Medical	Family and background query
Number of classes	1269	117
Number of sentences	9893	2393

grow and hence, not expected to distinguish performance improvements very well. Despite the use of agreement measure in the early stages of this work, because of its marginal benefit, the proposed methods have been mainly developed based on a second evaluation measure.

The second evaluation method is based on measuring the cluster purity, i.e., the average entropy of clusters. If $\mathcal{R}$ is the set of reference classes, the average entropy is defined for cluster set $\mathcal{C}$ as,

$$E = -\sum_{C\in\mathcal{C}} \frac{|C|}{|\mathcal{C}|} \sum_{R\in\mathcal{R}} P_{CR} \log(P_{CR}) \tag{9}$$

where,

$$P_{CR} \triangleq \frac{|C \cap R|}{\left|C \cap \left(\bigcup_{\Theta\in\mathcal{R}} \Theta\right)\right|} \tag{10}$$

Experimental results presented in Section 6 show that the cluster purity measure is more suitable for this task.

5.2. Overall evaluation

After training the classifier with the clustered data, an annotated test set can be used to measure the success level of the automatic training. To measure the classification accuracy, it suffices to count the number of cases that input utterance and the classifier output are from the same class in the reference annotations.

It is common in S2S translation systems to provide the user with multiple options. For instance in the system of Narayanan et al. (2003), the user is given a list of top four classifier outputs to choose from. In such cases, it is practically useful to measure the accuracy of the classifier within its n-best outputs (e.g., $n = 4$ for the above system). However, to avoid the accuracy bias, when that n-best list contains multiple correct answers, only one of them is counted.

6. Data and experiments

6.1. Classifier data

We used the two data sets shown in Table 1 to evaluate the performance of the proposed methods.

The first set was originally collected for, and used in, the Transonics project (Belvin et al., 2005) to develop an English/Farsi S2S translator in the doctor–patient interaction domain. For the doctor side (English), concept classes were carefully chosen using experts' judgment and medical phrase books (Narayanan et al., 2004). Then, for each concept, English data were collected from a website, a web-based game, and multiple paraphrasing sessions as explained in Section 2.

The second set was provided by *BBN Technologies* and consisted of paraphrases that were questions about peoples' family background.

6.2. SMT data

Two intermediate languages, Farsi and Iraqi Arabic, were chosen for this work. To train and optimize the statistical translation systems we used two parallel corpora.

The English/Farsi corpus was prepared for the Transonics project with the total size of 149,881 sentences (1,183,600 English running words). Most of the data were collected from general-domain conversation sessions and the rest were

Table 2
Classification accuracy for the conventional method and the proposed method with different lengths of *n*-best list.

	Conventional (baseline)	*n*-best length			
		100	500	1000	2000
Accuracy [%]	74.9	77.4	77.5	76.8	76.4
Relative error reduction [%]	–	10.0	10.4	7.6	6.0
Accuracy in 4-best [%]	88.6	90.7	91.0	91.3	90.5
Relative error reduction [%]	–	18.4	21.1	23.7	16.7

from interactions between medical students and actors performing as patients. Therefore this corpus was biased toward medical domain.

The second corpus was used in DARPA's *Transtac* project involving the development of an English/Arabic S2S translators (for more details see Choi et al. (2008)). This corpus consists of 654,181 sentences (5,517,656 English running words).

6.3. Sparsity handling

To compare the proposed method of Eq. (2) with the conventional classification (Eq. (1)), a classifier based on each method was put to test using the whole Transonics data set (Table 1). As the test corpus, 1000 phrases were randomly drawn from the above set and the rest were used for training. To make sure that the training set covered every class, one phrase per class was excluded from the test set selection process.

To generate the *n*-best lists, a phrase-based SMT (Koehn et al., 2003) was used. The intermediate language was Farsi and the SMT was trained on the parallel English/Farsi corpus with 147,691 sentences (1,168,856 English words) as we removed parts of the corpora that overlapped with the classifier data. This corpus was also used to build the classification background models in both languages. The SMT was optimized using a parallel development set with 915 lines (7281 English words).

In the proposed method, it is expected that the accuracy is affected by the length of the *n*-best lists. We used *n*-best lists of lengths 100, 500, 1000, and 2000 to observe that effect. The classification accuracy was measured on both the single output and the 4-best outputs. The results are shown in Table 2. In all of the above experiments the interpolation factor was set to 0.9 (weight 0.1 for background model) which is close to the optimum value obtained in Ettelaie et al. (2006).

To examine the effect of the background model, the conventional and proposed methods were tried with different values of the interpolation factor λ (the background model is weighted by $1-\lambda$). For the conventional method the length of the *n*-best list was set to 500. Fig. 2 shows the accuracy changes with respect to the interpolation factor for these two methods.

6.4. Clustering with n-best lists

First, we used the Transonics data by selecting 97 of the classes that contained at least four paraphrases. The associated 1207 English utterances were randomly split into 500 for training and 707 for testing with the assurance that each class at least had one sentence in the training set. To generate *n*-best lists, we trained and optimized the Moses system (Koehn et al., 2007) using the same English/Farsi data sets described in Section 6.3. The size of the lists were set to $n = 1000$.

The language models were generated using the SRILM toolkit (Stolcke, 2002). The KLD and JSD distance tables were formed by applying the algorithm from Sethy et al. (2004) to every pair of such language models. The distance tables were processed by both Exchange Method and Affinity Propagation algorithms to cluster the utterances. Throughout this work the number of classes was always considered a known parameter. When it is not known, this number can be determined by Affinity Propagation method. Here, by adjusting the preferences in Affinity Propagation, we fixed the number of clusters to the correct value. For each case, the Exchange Method was run 100,000 times with random initialization and the outcome that had the least cost (Eq. (5)) was selected.

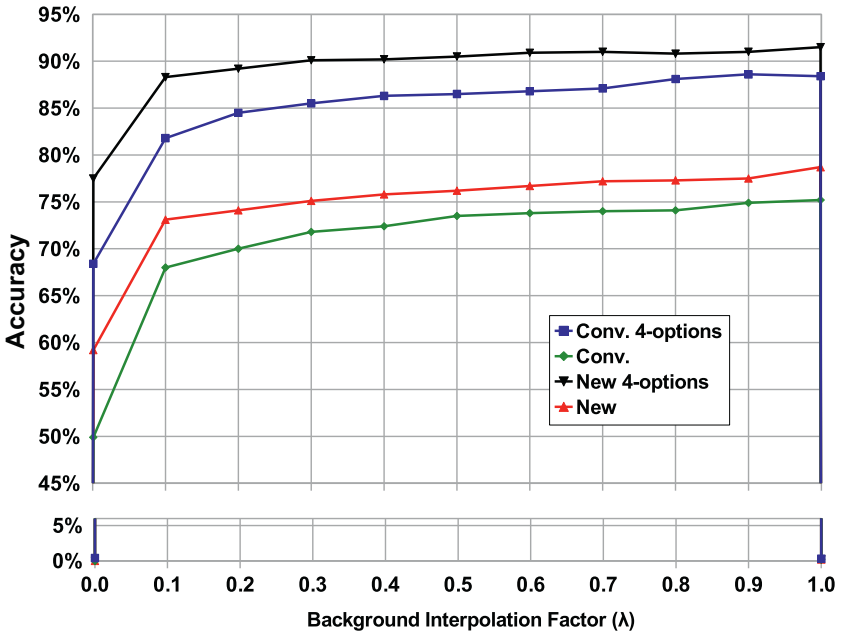

Fig. 2. The effect of background model on classification accuracy.

We also tried the spherical k-means algorithm using *gmeans* software (Dhillon et al., 2002). For that purpose, the *MC* toolkit (Dhillon et al., 2001) was first used to create the vector models from the documents (n-best lists). For comparison, k-means was applied to both the original English utterances and their associated n-best list documents (again $n = 1000$).

For comparison, random clustering in which the input utterances were randomly and uniformly dispersed over the 97 output clusters, was also included in the experiment. Table 3 shows the clustering agreement and cluster purity for the results of these different clustering methods along with the results of supervised training with annotated data. For the random clustering the table shows the average of measurements from 10,000 runs.

Since the main goal was to build a classifier, the outcome cluster from each method was used to train the classifier of Eq. (2) with $n = 50$. Table 3 also shows the classification accuracy and the accuracy within 4-best for each case which was measured using the testing data. For, all the cases we selected the class representatives using the method of Section 4.4. For k-means, random, and reference cases we used JSD distance tables in Eq. (7).

We repeated the same set of experiments on the BBN data set using all 117 classes. For training and testing 500 and 1000 sentences were randomly selected, with at least one sentence for each class in the training set.

Two intermediate languages were tried in this set of experiments. For Farsi, the Moses system was trained on 145,885 sentences (1,155,775 English words) from the English/Farsi corpus with 2000 sentences (15,872 English words) for development.

Table 3
The results of different clustering schemes for Transonics data.

Method	Agreement [%]	Purity [bits]	Acc. [%]	Acc. 4-best [%]
Random	97.47	3.780	14.43	35.08
Exchange Method with KLD ($n = 1000$)	**98.54**	**5.245**	**50.07**	**65.91**
Exchange Method with JSD ($n = 1000$)	98.39	5.106	46.11	63.08
Affinity Prop. with KLD ($n = 1000$)	98.33	5.099	49.36	65.63
Affinity Prop. with JSD ($n = 1000$)	98.33	5.077	47.95	62.66
Spherical k-means with original data	97.92	4.763	38.61	54.60
Spherical k-means with n-best documents	98.06	4.890	42.01	55.16
Reference annotation	100.0	6.213	72.84	87.98

Table 4
The results of different clustering schemes for BBN data.

Method	Intermediate language	Agr. [%]	Purity [bits]	Acc. [%]	Acc. 4-best [%]
Random	–	98.07	4.334	13.20	29.10
Exchange Method with KLD ($n = 1000$)	Farsi	98.51	5.240	37.50	54.20
	Arabic	98.55	5.269	34.90	53.70
Exchange Method with JSD ($n = 1000$)	Farsi	98.36	5.031	31.40	48.20
	Arabic	98.31	4.989	31.60	49.50
Affinity Propagation with KLD ($n = 1000$)	Farsi	**98.63**	**5.350**	**40.50**	**57.50**
	Arabic	98.62	5.327	39.10	56.20
Affinity Propagation with JSD ($n = 1000$)	Farsi	98.46	5.099	33.70	52.30
	Arabic	98.39	5.086	32.10	50.60
Spherical k-means with original data	English	98.40	5.264	33.20	51.70
Spherical k-means with n-best documents	Farsi	98.54	5.321	36.30	54.70
Reference annotation	–	100.0	6.561	60.20	77.90

To train the Moses system for English to Arabic translation, the Transtac corpus was split into a 651,181 sentence (5,486,547 English words) training set and a 2000 sentence (22,632 English words) development set.

The results of the experiments on the BBN data are shown in Table 4. In all the cases, the classification of the test set was performed with lists of size $n' = 50$ in Farsi (Eq. (2)).

6.5. The effect of n-best lists size

To study the effect of the n-best lists size on the clustering process, we used the same setup explained in the previous section for Transonics data and generated n-best lists of 50, 100, 500, 1000, 2000, and 3000 hypotheses per source sentence. Then for each size, the KLD distance table was generated. The Exchange Method and the Affinity Propagation were carried out using these tables and the clustering agreement and cluster purity were measured for each case. These measurements are reported in Table 5 along with the corresponding classifier accuracy for the testing data.

7. Results and discussion

7.1. Classification by n-best lists

Table 5 shows the advantage of the proposed method over the conventional classification with a relative error rate reduction up to 10.4% (achieved when the length of the SMT n-best list was 500). However, as expected, this number decreases with longer SMT n-best lists due to the increased noise present in lower ranked outputs of the SMT.

Table 5 also shows the accuracy within 4-best classifier outputs for each method. In that case the proposed method showed an error rate which was relatively 23.7% lower than the error rate of the conventional method. That was

Table 5
The effect of the size of the SMT n-best list using Exchange Method and Affinity Propagation with KLD metric.

Method	n-best length	50	100	500	1000	2000	3000
Exchange Method	Agreement [%]	98.48	98.56	**98.63**	98.54	98.60	98.55
	Cluster purity [bits]	5.222	5.281	**5.323**	5.245	5.308	5.273
	Accuracy [%]	45.40	50.21	**51.63**	50.07	50.92	48.51
	Acc. within 4-best [%]	59.97	63.65	**66.62**	65.91	65.06	64.22
Affinity Prop.	Agreement [%]	**98.33**	98.28	**98.33**	**98.33**	98.31	98.32
	Cluster purity [bits]	**5.100**	5.079	5.091	5.099	5.080	5.097
	Accuracy [%]	49.22	48.51	48.94	**49.36**	47.81	48.66
	Acc. within 4-best [%]	65.06	63.65	64.22	65.63	65.06	**68.03**

achieved at the peak of the accuracy within 4-best, when the length of the SMT *n*-best list was 1000. In this case too, further increase in the length of the *n*-best list led to an accuracy degradation as the classifier models became noisier.

7.2. Background model

The effect of the background model on classifier accuracy is shown in Fig. 2. The figure shows the one-best accuracy and the accuracy within 4-best outputs, versus the interpolation factor (λ) for both conventional and proposed methods. As the curves indicate, with λ equal to zero the classifier has no discriminating feature since all the class scores are driven solely from the background model. However, a slight increase in λ, leads to a large jump in the accuracy. The reason is that the background model was built from a large general domain corpus and hence, had no bias toward any of the classes. With a small λ, the score from the background model dominates the overall class scores. In spite of that, the score differences caused by the class LM's are notable in improving the classifier performance.

As λ increases the role of the class LM's becomes more prominent. This makes the classifier models more discriminative and increases its accuracy as shown in Fig. 1. When the factor is in the close vicinity of one, the smoothing effect of the background model diminishes and leaves the classes with spiky models with very low vocabulary coverage (lots of zeros). This leads to a rapid drop in accuracy as λ reaches one.

Both the conventional and proposed methods follow the above trend as Fig. 2 shows, although, the proposed method maintains its superiority throughout the range of λ that was examined. The maximum measured accuracy numbers for conventional and proposed methods were 75.2% and 78.7% respectively and was measured at $\lambda = 0.999$ for both methods. Therefore, the error rate of the proposed method was relatively 14.1% lower than its counterpart from the conventional method.

Fig. 2 also indicates that when the accuracy is measured within the 4-best outputs, again the proposed method outperforms the conventional one. The maximum 4-best accuracy for the conventional method was measured at the sample point $\lambda = 0.9$ and was equal to 88.6%. For the proposed method, that number was measured as 91.5% achieved at the sample point $\lambda = 0.999$. In another words, considering the 4-best classifier outputs, the error rate of the proposed method was relatively 25.4% lower.

7.3. Clustering with new metrics

The results in Tables 3 and 4 provide insights that can help the further development of an unsupervised training method. The cluster purity and the agreement are included to show the quality of the clustering process in each case. These numbers indicate that, with some exceptions, a better clustering leads to a more accurate classifier. Although, for a rigorous proof a statistical analysis on the outcome of numerous experiments is necessary. The cluster purity seems to be a more useful measure for evaluating the quality of the clusters as the other measure saturates rapidly.

The tables clearly show the superiority of methods that use SMT *n*-best list for distance calculation. Using *n*-best lists (Farsi as intermediate language with $n = 1000$) in *k*-means algorithm improved the classification accuracy by about 9% (relatively) for both Transonics and BBN data sets.

As it is clear from Tables 3 and 4, the clustering methods with information theoretic distance have an advantage over the *k*-means algorithm which is based on vector representation and Euclidean distance. In both data sets, KLD yielded better results in terms of both clustering quality and classification accuracy that followed it. Since JSD is a smoother measure, it is less likely to capture the small LM differences that are crucial in the clustering process.

For the Transonics data, the Exchange Method gave the best results by all measures. The classifier accuracy in that case was relatively 30% better than the accuracy from the spherical *k*-means. With the same methods and the same intermediate language, this accuracy improvement was 13% for BBN data. For the latter data set, the best results by all measures were achieved by Affinity Propagation clustering with Farsi as the intermediate language. In that case, a 22% relative increase over the accuracy resulted from *k*-means was observed. A better result for Transonics data was expected because of the match to the SMT domain (medical). It is worth mentioning that the result of our experiments with French and Spanish as the intermediate languages, using *Europarl* corpora, were inferior due to the domain mismatch.

For BBN data, using Farsi and Arabic as the intermediate language produced very close results (except for Exchange Method with KLD), although Farsi produced a slightly better numbers. The Arabic corpus used to train the SMT was

more than four times larger than its Farsi counterpart, however Arabic as a highly inflected language seemed to be a less efficient choice as an intermediate language.

In both data sets, the utterances are not evenly distributed over the classes as some of the concepts, e.g., greetings are much more frequent than the others. The original distribution is more or less preserved while sampling the training and the testing sets and leaves some classes with more items. With random clustering, these items dominate some of the clusters. While testing, the items from more frequent classes are classified to these dominated clusters and labeled correctly. Therefore even with random clustering the average accuracy was 14.43% and 13.20% for Transonics and BBN data sets.

7.4. Clustering and the size of the n-best lists

Table 5 shows how the size of the n-best list affects the clustering process. To decouple the size impact on clustering and classification, in all the cases the classifier was built using the n-best list with size $n = 50$.

Clustering with Exchange Method seemed to follow a more expected pattern. An increase in accuracy was observed as the size of the n-best lists grew from 50 to 500 for which an accuracy of 51.63% was achieved. We see a decline as the lists grow larger. As more SMT hypotheses are included in the distance measurements, a better clustering is achieved due to the lexical diversity. However, for larger lists the assumption that all the hypotheses are quality translations of the source sentence, loses its validity. Low quality hypotheses in the bottom of the list increases the noise in the distance table and cause an inferior clustering result. Consequently there is a trade-off between lexical enhancement versus noise control. The same trend can be observed for accuracy within 4-best and more or less for the cluster purity.

The results of the Affinity Propagation algorithm did not follow the same trend. Although the peaking effect happened in that case too. Since the algorithm uses the original KLD as the distance measure it is more sensitive to the distance errors. In fact, Eq. (3) can be viewed as a mild smoothing process which gives some level of noise tolerance to the Exchange Method that uses the symmetric KLD. As Table 5 shows the accuracy peak with Affinity Propagation clustering has occurred for $n = 1000$.

8. Conclusions

By using the n-best lists of imperfect translations in the proposed method the accuracy of the concept classifiers can be improved, especially when the training data are sparse. As experiments showed, our method outperformed the conventional classifier, trained on the original source language paraphrases. When the input utterance is within the classification domain, the proposed method can be viewed as a filtering technique that produces fluent translations (removes the "noise") from the SMT output.

We believe that further improvements to the technique can be made through the use of weighted n-best lists based on the SMT scores. In addition using a much richer SMT engine could provide significant gains through increased diversity in the output vocabulary. We intend to extend on this work through the use of multiple classifiers in different intermediate languages for which there are available resources to build enriched SMT engines.

The experiments also emphasized the importance of the background model and indicated that, with a proper selection of such model, the classification accuracy was not very sensitive to the value of the interpolation factor. This eliminates the tuning process and the need for development data.

Clustering the data utterances based on their concept is necessary for unsupervised training of a classifier-based translation system. The main focus of this work was to propose a distance measure to compare utterances based on their concepts. The utterances are compared based on their multiple translations. To generate these translations, n-best lists from an SMT system are used and language models are generated from these lists. Information theoretic metrics are used to quantify the difference between these language models.

We also showed that the Exchange Method and the Affinity Propagation algorithms could be adapted as an appropriate clustering method to use with these metrics. The n-best lists can also be used directly in the k-means algorithm. The effectiveness of these methods were compared through a set of experiments.

In experiments with two different data sets, clustering with the proposed metrics showed superior results compared to the classic k-means. The classifiers trained on the outcome of these clustering tasks also showed more than 20% (up to 30%) higher accuracy compared to the one trained on the outputs of k-means. For accuracy within 4-best, the improvement was at least 11% and up to 21%.

The effect of the size of the SMT generated *n*-best lists was also examined. The experiments show that the classifier accuracy peaks and then drops as the *n*-best list length grows.

By the proposed methods, the concept classifier can be trained automatically. However, as expected, the classification accuracy would be significantly lower than what could be achieved by supervised training. This work is the first step in the development of unsupervised training methods for classifier based translation systems. We are currently focused on improving the quality of the clusters through different methods including filtering in different stages of the process, such as, original data selection, *n*-best list generation, and clustering.

Acknowledgments

We would like to thank BBN Technologies for sharing their data with us and Tom Murray for his help with *gmeans* software. This work was supported in part by funds from DARPA and NSF. The computing facilities were provided by USC's High-Performance Computing and Communications Center.

References

Belvin, R., Ettelaie, E., Gandhe, S., Georgiou, P., Knight, K., Marcu, M., Millward, S., Narayanan, S., Neely, H., Traum, D., 2005. Transonics: a practical speech-to-speech translator for English–Farsi medical dialogs. In: Proc. of the Association for Computational Linguistics, Interactive Poster and Demonstration Sessions, Ann Arbor, MI, USA, pp. 89–92.

Chklovski, T., 2005. Collecting paraphrase corpora from volunteer contributors. In: Proc. of the Third International Conference on Knowledge Capture (K-CAP), Banff, Canada, pp. 115–120.

Choi, F., Tsakalidis, S., Saleem, S., Lin Kao, C., Meermeier, R., Krstovski, K., Moran, C., Subramanian, K., Prasad, R., Natarajan, P., 2008. Recent improvements in BBN's English/Iraqi speech-to-speech translation system. In: Proc. of the Second IEEE Workshop on Spoken Language Technology (SLT), Goa, India, pp. 245–248.

Dhillon, I.S., Modha, D.S., 2001. Concept decompositions for large sparse text data using clustering. Machine Learning 42 (1), 143–175.

Dhillon, I.S., Fan, J., Guan, Y., 2001. Efficient clustering of very large document collections. In: Grossman, V.K.R., Kamath, C., Namburu, R. (Eds.), Data Mining for Scientific and Engineering Applications. Kluwer Academic Publishers, pp. 357–381.

Dhillon, I.S., Guan, Y., Kogan, J., 2002. Iterative clustering of high dimensional text data augmented by local search. In: Proc. of the IEEE International Conference on Data Mining (ICDM), Maebashi City, Japan, pp. 131–138.

Ehsani, F., Kinzey, J., Master, D., Sudre, K., Domingo, D., Park, H., 2006. S-MINDS 2-way speech-to-speech translation system. In: Proc. of the Medical Speech Translation Workshop, Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT), New York, NY, USA, pp. 44–45.

Ettelaie, E., Georgiou, P.G., Narayanan, S., 2006. Cross-lingual dialog model for speech to speech translation. In: Proc. of the Ninth International Conference on Spoken Language Processing (ICSLP), Pittsburgh, PA, USA, pp. 1173–1176.

Ettelaie, E., Georgiou, P.G., Narayanan, S., 2008. Towards unsupervised training of the classifier-based speech translator. In: Proc. of the International Conference on Spoken Language Processing (ICSLP), Brisbane, Australia, pp. 2739–2742.

Ettelaie, E., Georgiou, P.G., Narayanan, S., 2008. Mitigation of data sparsity in classifier-based translation. In: Proc. of Coling 2008 Workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, Manchester, UK, pp. 1–4.

Ettelaie, E., Georgiou, P.G., Narayanan, S., 2010. Hierarchical classification for speech-to-speech translation. In: Proc. of the Interspeech, Makuhari, Japan.

Frey, B.J., Dueck, D., 2007. Clustering by passing messages between data points. Science 315, 972–976.

Gao, Y., Gu, L., Zhou, B., Sarikaya, R., Afify, M., Kuo, H., Zhu, W., Deng, Y., Prosser, C., Zhang, W., Besacier, L., 2006. IBM MASTOR SYSTEM: multilingual automatic speech-to-speech translator. In: Proc. of the Medical Speech Translation Workshop, Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT), New York, NY, USA, pp. 53–56.

Hatzivassiloglou, V., Klavans, J., Holcombe, M., Barzilay, R., Kan, M., McKeown, K., 2001. Simfinder: a flexible clustering tool for summarization. In: Proc. of NAACL Workshop on Automatic Summarization, Pittsburgh, PA, USA, pp. 41–49.

Hsiao, R., Venugopal, A., Kohler, T., Zhang, Y., Charoenpornsawat, P., Zollmann, A., Vogel, S., Black, A.W., Schultz, T., Waibel, A., 2006. Optimizing components for handheld two-way speech translation for an English–Iraqi Arabic system. In: Proc. of the Ninth International Conference on Spoken Language Processing (ICSLP), Pittsburgh, PA, USA, pp. 765–768.

Koehn, P., Och, F., Marcu, D., 2003. Statistical phrase-based translation. In: Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT), vol. 1, Edmonton, AB, Canada, pp. 48–54.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E., 2007. Moses: open source toolkit for statistical machine translation. In: Proc. of the 45th Annual Meeting of the Association for Computational Linguistics (ACL), Vol. Companion Proc. of the Demo and Poster Sessions, Prague, Czech Republic, pp. 177–180.

Kschischang, F., Frey, B.J., Loeliger, H.-A., 2001. Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory 47, 498–519.

Leuski, A., Pair, J., Traum, D., McNerney, P.J., Georgiou, P., Patel, R., 2006. How to talk to a hologram. In: Proc. of the Eleventh International Conference on Intelligent User Interfaces (IUI), Sydney, Australia, pp. 360–362.

Lin, J., 1991. Divergence measures based on the Shannon entropy. IEEE Transactions on Information Theory 37 (1), 145–151.

McKeown, K., Hatzivassiloglou, V., 1993. Augmenting lexicons automatically: clustering semantically related adjectives. In: Proc. of ARPA Workshop on Human Language Technology (HLT'93), Princeton, NJ, USA, pp. 272–277.

Narayanan, S., Ananthakrishnan, S., Belvin, R., Ettelaie, E., Ganjavi, S., Georgiou, P., Hein, C., Kadambe, S., Knight, K., Marcu, D., Neely, H., Srinivasamurthy, N., Traum, D., Wang, D., 2003. Transonics: a speech to speech system for English–Persian interactions. In: Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), St. Thomas, US Virgin Islands, pp. 670–675.

Narayanan, S., Ananthakrishnan, S., Belvin, R., Ettelaie, E., Gandhe, S., Ganjavi, S., Georgiou, P.G., Hein, C.M., Kadambe, S., Knight, K., Marcu, D., Neely, H.E., Srinivasamurthy, N., Traum, D., Wang, D., 2004. The transonics spoken dialogue translator: an aid for English–Persian doctor–patient interviews. In: Proc. of American Association for Artificial Intelligence Fall Symposium on Dialog Systems for Health Communication (AAAI), Arlington, VA.

Narayanan, S.S., Georgiou, P.G., Sethy, A., Wang, D., Bulut, M., Sundaram, S., Ettalaie, E., Ananthakrishnan, S., Franco, H., Precoda, K., Vergyri, D., Zheng, J., Wang, W., Gadde, R.R., Graciarena, M., Abrash, V., Frandsen, M., Richey, C., 2006. Speech recognition engineering issues in speech to speech translation system design for low resource languages and domains. In: Proc. of the Thirty First IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 5, Toulose, France, pp. 1209–1212.

Ney, H., Nießen, S., Och, F.J., Tillmann, C., Sawaf, H., Vogel, S., 2000. Algorithms for statistical translation of spoken language. IEEE Transaction on Speech and Audio Processing, Special Issue on Language Modeling and Dialogue Systems 8 (1), 24–36.

Och, F., Ney, H., 2004. The alignment template approach to statistical machine translation. Computational Linguistics 30 (4), 417–449.

Papineni, K., Roukos, S., Ward, T., Zhu, W., 2012. Bleu: a method for automatic evaluation of machine translation, Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center.

Späth, H., 1985. The Cluster Dissection and Analysis Theory FORTRAN Programs Examples. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Sethy, A., Narayanan, S., Ramabhadran, B., 2004. Measuring convergence in language model estimation using relative entropy. In: Proc. of the Eight International Conference on Spoken Language Processing (ICSLP), Jeju Island, Korea, pp. 1057–1060.

Stallard, D., Choi, F., Krstovski, K., Natarajan, P., Prasad, R., Saleem, S., 2006. A hybrid phrase-based/statistical speech translation system. In: Proc. of the Ninth International Conference on Spoken Language Processing (ICSLP), Pittsburgh, PA, USA, pp. 757–760.

Stolcke, A., 2002. SRILM – an extensible language modeling toolkit. In: Proc. of the International Conference on Spoken Language Processing (ICSLP), Denver, CO, USA, pp. 901–904.

Traum, D., Roque, A., Leuski, A., Georgiou, P., Gerten, J., Martinovski, B., Narayanan, S., Robinson, S., Vaswani, A., 2007. Hassan: a virtual human for tactical questioning. In: Proc. of the Eighth SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium, pp. 75–78.

Wagstaff, K., Cardie, C.,2000. Clustering with instance-level constraints. In: Proc. of the Seventeenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1103–1110.

Ye, H., Young, S., 2006. A clustering approach to semantic decoding. In: Proc. of the Ninth International Conference on Spoken Language Processing (ICSLP), Pittsburgh, PA, USA, pp. 5–8.