Abstract

We invent referential translation machines (RTMs), a computational model for identifying the translation acts between any two data sets with respect to a reference corpus selected in the same domain, which can be used for judging the semantic similarity between text. RTMs make quality and semantic similarity judgments possible by using retrieved relevant training data as interpretants for reaching shared semantics. An MTPP (machine translation performance predictor) model derives features measuring the closeness of the test sentences to the training data, the difficulty of translating them, and the presence of acts of translation involved. We view semantic similarity as paraphrasing between any two given texts. Each view is modeled by an RTM model, giving us a new perspective on the binary relationship between the two. Our prediction model is the 15th on some tasks and 30th overall out of 89 submissions in total according to the official results of the Semantic Textual Similarity (STS 2013) challenge.

1 Semantic Textual Similarity Judgments

We introduce a fully automated judge for semantic similarity that performs well in the semantic textual similarity (STS) task (Agirre et al., 2013). STS is a degree of semantic equivalence between two texts based on the observations that “vehicle” and “car” are more similar than “wave” and “car”. Accurate prediction of STS has a wide application area including: identifying whether two tweets are talking about the same thing, whether an answer is correct by comparing it with a reference answer, and whether a given shorter text is a valid summary of another text.

The translation quality estimation task (Callison-Burch et al., 2012) aims to develop quality indicators for translations at the sentence-level and predictors without access to a reference translation. Bicici et al. (2013) develop a top performing machine translation performance predictor (MTPP), which uses machine learning models over features measuring how well the test set matches the training set relying on extrinsic and language independent features.

The semantic textual similarity (STS) task (Agirre et al., 2013) addresses the following problem. Given two sentences $S_1$ and $S_2$ in the same language, quantify the degree of similarity with a similarity score, which is a number in the range $[0, 5]$. The semantic textual similarity prediction problem involves finding a function $f$ approximating the semantic textual similarity score given two sentences, $S_1$ and $S_2$:

$$f(S_1, S_2) \approx q(S_1, S_2).$$

(1)

We approach $f$ as a supervised learning problem with $(S_1, S_2, q(S_1, S_2))$ tuples being the training data and $q(S_1, S_2)$ being the target similarity score.

We model the problem as a translation task where one possible interpretation is obtained by translating $S_1$ (the source to translate, S) to $S_2$ (the target translation, T). Since linguistic processing can reveal deeper similarity relationships, we also look at the translation task at different granularities of information: plain text (R for regular), after lemmatization (L), after part-of-speech (POS) tagging (P), and after removing 128 English stop-words (S)\(^1\). Thus,

\(^1\)http://anoncvs.postgresql.org/cvsweb.cgi/pgsql/
we obtain 4 different perspectives on the binary relationship between $S_1$ and $S_2$.

2 Referential Translation Machine (RTM)

Referential translation machines (RTMs) we develop provide a computational model for quality and semantic similarity judgments using retrieval of relevant training data (Biçici and Yuret, 2011a; Biçici, 2011) as interpretants for reaching shared semantics (Biçici, 2008). We show that RTM achieves very good performance in judging the semantic similarity of sentences and we can also use RTM to automatically assess the correctness of student answers to obtain better results (Biçici and van Genabith, 2013) than the state-of-the-art (Dzikovska et al., 2012).

RTM is a computational model for identifying the acts of translation for translating between any given two data sets with respect to a reference corpus selected in the same domain. RTM can be used for automatically judging the semantic similarity between texts. An RTM model is based on the selection of common training data relevant and close to both the training set and the test set where the selected relevant set of instances are called the interpretants. Interpretants allow shared semantics to be possible by behaving as a reference point for similarity judgments and providing the context. In semiotics, an interpretant $I$ interprets the signs used to refer to the real objects (Biçici, 2008). RTMs provide a model for computational semantics using interpretants as a reference according to which semantic judgments with translation acts are made. Each RTM model is a data translation model between the instances in the training set and the test set. We use the FDA (Feature Decay Algorithms) instance selection model for selecting the interpretants (Biçici and Yuret, 2011a) from a given corpus, which can be monolingual when modeling paraphrasing acts, in which case the MTPP model (Section 2.1) is built using the interpretants themselves as both the source and the target side of the parallel corpus. RTMs map the training and test data to a space where translation acts can be identified. We view that acts of translation are ubiquitously used during communication:

*Every act of communication is an act of translation* (Bliss, 2012).

Translation need not be between different languages and paraphrasing or communication also contain acts of translation. When creating sentences, we use our background knowledge and translate information content according to the current context.

Given a training set $\text{train}$, a test set $\text{test}$, and some monolingual corpus $C$, preferably in the same domain as the training and test sets, the RTM steps are:

1. $T = \text{train} \cup \text{test}$. 
2. select($T, C$) $\rightarrow I$
3. $\text{MTPP}(I, \text{train}) \rightarrow F_{\text{train}}$
4. $\text{MTPP}(I, \text{test}) \rightarrow F_{\text{test}}$
5. learn($M, F_{\text{train}}$) $\rightarrow M$
6. $\text{predict}(M, F_{\text{test}}) \rightarrow \hat{q}$

Step 2 selects the interpretants, $I$, relevant to the instances in the combined training and test data. Steps 3, 4 use $I$ to map $\text{train}$ and $\text{test}$ to a new space where similarities between translation acts can be derived more easily. Step 5 trains a learning model $M$ over the training features, $F_{\text{train}}$, and Step 6 obtains the predictions. RTM relies on the representativeness of $I$ as a medium for building translation models for translating between $\text{train}$ and $\text{test}$.

Our encouraging results in the STS task provides a greater understanding of the acts of translation we ubiquitously use when communicating and how they can be used to predict the performance of translation, judging the semantic similarity between text, and evaluating the quality of student answers. RTM and MTPP models are not data or language specific and their modeling power and good performance are applicable across different domains and tasks. RTM expands the applicability of MTPP by making it feasible when making monolingual quality and similarity judgments and it enhances the computational scalibility by building models over smaller but more relevant training data as interpretants.

2.1 The Machine Translation Performance Predictor (MTPP)

In machine translation (MT), pairs of source and target sentences are used for training statistical MT (SMT) models. SMT system performance is affected by the amount of training data used as well
as the closeness of the test set to the training set. MTPP (Biçici et al., 2013) is a top performing machine translation performance predictor, which uses machine learning models over features measuring how well the test set matches the training set to predict the quality of a translation without using a reference translation. MTPP measures the coverage of individual test sentence features and syntactic structures found in the training set and derives feature functions measuring the closeness of test sentences to the available training data, the difficulty of translating the sentence, and the presence of acts of translation for data transformation.

2.2 MTPP Features for Translation Acts

MTPP uses $n$-gram features defined over text or common cover link (CCL) (Seginer, 2007) structures as the basic units of information over which similarity calculations are made. Unsupervised parsing with CCL extracts links from base words to head words, which allow us to obtain structures representing the grammatical information instantiated in the training and test data. Feature functions use statistics involving the training set and the test sentences to determine their closeness. Since they are language independent, MTPP allows quality estimation to be performed extrinsically. Categories for the 289 features used are listed below and their detailed descriptions are presented in (Biçici et al., 2013) where the number of features are given in {#}.

- **Coverage** {110}: Measures the degree to which the test features are found in the training set for both $S$ ({56}) and $T$ ({54}).
- **Synthetic Translation Performance** {6}: Calculates translation scores achievable according to the $n$-gram coverage.
- **Length** {4}: Calculates the number of words and characters for $S$ and $T$ and their ratios.
- **Feature Vector Similarity** {16}: Calculates the similarities between vector representations.
- **Perplexity** {90}: Measures the fluency of the sentences according to language models (LM). We use both forward ({30}) and backward ({15}) LM based features for $S$ and $T$.
- **Entropy** {4}: Calculates the distributional similarity of test sentences to the training set.
- **Retrieval Closeness** {24}: Measures the degree to which sentences close to the test set are found in the training set.
- **Diversity** {6}: Measures the diversity of co-occurring features in the training set.
- **IBM1 Translation Probability** {16}: Calculates the translation probability of test sentences using the training set (Brown et al., 1993).
- **Minimum Bayes Retrieval Risk** {4}: Calculates the translation probability for the translation having the minimum Bayes risk among the retrieved training instances.
- **Sentence Translation Performance** {3}: Calculates translation scores obtained according to $q(T, R)$ using BLEU (Papineni et al., 2002), NIST (Doddington, 2002), or $F_1$ (Biçici and Yuret, 2011b) for $q$.
- **Character $n$-grams** {4}: Calculates the cosine similarities between the character $n$-grams (for $n=2,3,4,5$) obtained for $S$ and $T$ (Bår et al., 2012).
- **LIX** {2}: Calculates the LIX readability score (Wikipedia, 2013; Björnsson, 1968) for $S$ and $T$. \(^2\)

3 Experiments

STS contains sentence pairs from news headlines (headlines), sense definitions from semantic lexical resources (OnWN is from OntoNotes (Pradhan et al., 2007) and WordNet (Miller, 1995) and FNWN is from FrameNet (Baker et al., 1998) and WordNet), and statistical machine translation (SMT) (Agirre et al., 2013). STS challenge results are evaluated with the Pearson’s correlation score ($r$).

The test set contains 2250 ($S_1$, $S_2$) sentence pairs with 750, 561, 189, and 750 sentences from each type respectively. The training set contains 5342 sentence pairs with 1500 each from MSRpar and MSRvid (Microsoft Research paraphrase and video description corpus (Agirre et al., 2012)), 1592 from SMT, and 750 from OnWN.

3.1 RTM Models

We obtain CNGL results for the STS task as follows. For each perspective described in Section 1, we build an RTM model. Each RTM model views the STS task from a different perspective using the 289 features extracted dependent on the interpreters using MTPP. We extract the features both on

\[ LIX = \frac{A}{B} + C \frac{100}{D}, \]

where $A$ is the number of words, $C$ is words longer than 6 characters, $B$ is words that start or end with any of “.”, “:”, “!”, “?” similar to (Hagström, 2012).
the training set and the test set. The training corpus used is the English side of an out-of-domain corpus on European parliamentary discussions, Europarl (Callison-Burch et al., 2012). In-domain corpora are likely to improve the performance. We use the Stanford POS tagger (Toutanova et al., 2003) to obtain the perspectives P and L. We use the training corpus to build a 5-gram target LM.

We use ridge regression (RR) and support vector regression (SVR) with RBF kernel (Smola and Schölkopf, 2004). Both of these models learn a regression function using the features to estimate a numerical target value. The parameters that govern the behavior of RR and SVR are the regularization λ for RR and the C, ϵ, and γ parameters for SVR. At testing time, the predictions are bounded to obtain scores in the range [0, 5]. We perform tuning on a subset of the training set separately for each RTM model and optimize against the performance evaluated with $R^2$, the coefficient of determination.

We do not build a separate model for different types of sentences and instead use all of the training set for building a large prediction model. We also use transductive learning since using only the relevant training data for training can improve the performance (Biçici, 2011). Transductive learning is performed at the sentence level where for each test instance, we select 1250 relevant training instances using the cosine similarity metric over the feature vectors and build an individual model for the test instance and predict the similarity score.

### 3.2 Training Results

Table 1 lists the 10-fold cross-validation (CV) results on the training set for RR and SVR for different RTM systems using optimized parameters. As we combine different perspectives, the performance improves and we use the L+S with SVR for run 1 (LSSVR), L+P+S with SVR for run 2 (LPSSVR), and L+P+S with SVR using transductive learning for run 3 (LPSSVRTL) all in the translation direction $S_1 \rightarrow S_2$. Lemmatized RTM, L, performs the best among the individual perspectives. We also build RTM models in the direction $S_2 \rightarrow S_1$, which gives similar results. The last main row combines them to obtain the bi-directional results, $S_1 \rightleftarrows S_2$, which improves the performance. Each additional perspective adds another 289 features to the representation and the bi-directional results double the number of features. Thus, $S_1 \rightleftarrows S_2$ L+P+S is using 1734 features.

### 3.3 STS Challenge Results

Table 2 presents the STS challenge $r$ and ranking results containing our CNGL submissions, the best system result, and the mean results over all submissions. There were 89 submissions from 35 competing systems (Agirre et al., 2013). The results are ranked according to the mean $r$ obtained. We also include the mean result over all of the submissions and its corresponding rank.

According to the official results, CNGL-LSSVR is the 30th system from the top based on the mean $r$ obtained and CNGL-LPSSVR is 15th according to the results on OnWN out of 89 submissions in total.
CNGL submissions perform unexpectedly low in the FNWN task and only slightly better than the average in the SMT task. The lower performance is likely to be due to using an out-of-domain corpus for building the RTM models and it may also be due to using and optimizing a single model for all types of tasks.

### 3.4 Bi-directional RTM Models

The STS task similarity score is directional invariant: \( q(S_1, S_2) = q(S_2, S_1) \). We develop RTM models in the reverse direction and obtain bi-directional RTM models by combining both. Table 3 lists the bi-directional results on the STS challenge test set after tuning, which shows that slight improvement in the scores are possible when compared with Table 2. Transductive learning improves the performance in general. We also compare with the performance obtained when combining unidirectional models with mean, min, or max functions. Taking the minimum performs better than other combination approaches and can achieve \( r = 0.5129 \) with TL. One can also take the individual confidence scores obtained for each score when combining scores.

### 4 Conclusion

Referential translation machines provide a clean and intuitive computational model for automatically measuring semantic similarity by measuring the acts of translation involved and achieve to be the 15th on some tasks and 30th overall in the STS challenge out of 89 submissions in total. RTMs make quality and semantic similarity judgments possible based on the retrieval of relevant training data as interpreters for reaching shared semantics.


Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech