CNGL-CORE: Referential Translation Machines for Measuring Semantic Similarity

Ergun Biçici
Centre for Next Generation Localisation,
Dublin City University, Dublin, Ireland.
ebicici@computing.dcu.ie

Josef van Genabith
Centre for Next Generation Localisation,
Dublin City University, Dublin, Ireland.
josef@computing.dcu.ie

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:

$$f(S_1, S_2) \approx q(S_1, S_2). \tag{1}$$

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

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

Step 2 selects the interpretants, $\mathcal{I}$, relevant to the instances in the combined training and test data. Steps 3, 4 use $\mathcal{I}$ to map `train` and `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, $\mathcal{F}_{\texttt{train}}$, and Step 6 obtains the predictions. RTM relies on the representativeness of $\mathcal{I}$ as a medium for building translation models for translating between `train` and `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 scalability 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

src/backend/snowball/stopwords/

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 between the character n-grams (for n=2,3,4,5) obtained for S and T (Bär 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 interpretants using MTPP. We extract the features both on

[2] LIX=$\frac{A}{B} + C\frac{100}{A}$, 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).

	r	R	P	L	S	R+P	R+L	R+S	L+P	L+S	L+S TL	R+P+L	R+P+S	L+P+S	L+P+S TL	R+P+L+S	R+P+L+S TL
$S_1 \rightarrow S_2$	RR	.7904	.7502	.8200	.7788	.8074	.8232	.8101	.8247	.8218	.8509	.8266	.8172	.8304	.8530	.8323	.8499
	SVR	.8311	.8060	.8443	.8330	.8404	.8517	.8498	.8501	.8593	.8556	.8496	.8422	.8586	.8579	.8527	.8564
$S_2 \rightarrow S_1$	RR	.7922	.7651	.8169	.7891	.8064	.8196	.8136	.8219	.8257	.8257	.8226	.8164	.8284	.8284	.8313	.8324
	SVR	.8308	.8165	.8407	.8302	.8361	.8506	.8467	.8510	.8567	.8567	.8525	.8460	.8588	.8588	.8575	.8574
$S_1 \rightleftarrows S_2$	RR	.8079	.787	.8279	.8101	.8216	.8333	.8275	.8346	.8375	.8409	.8361	.8312	.8412	.8434	.8432	.844
	SVR	.8397	.8237	.8554	.841	.8432	.857	.851	.8557	.8605	.8626	.8505	.8505	.8591	.8622	.8602	.8588

Table 1: CV performance on the training set with tuning. Underlined are the settings we use in our submissions. RTM models in directions $S_1 \rightarrow S_2$, $S_2 \rightarrow S_1$, and the bi-directional models $S_1 \rightleftarrows S_2$ are displayed.

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) [3]. 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.

[3] We use WMT'13 corpora from www.statmt.org/wmt13/.

System	head	OnWN	FNWN	SMT	mean	rank
CNGL-LSSVR	.6552	.6943	.2016	.3005	.5086	30
CNGL-LPSSVRTL	.6385	.6756	.1823	.3098	.4998	33
CNGL-LPSSVR	.6510	.6971	.1180	.2861	.4961	36
UMBC-EB.-PW	.7642	.7529	.5818	.3804	.6181	1
mean	.6071	.5089	.2906	.3004	.4538	57

Table 2: STS challenge r and ranking results ranked according to the mean r obtained. head is headlines and mean is the mean of all submissions.

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 uni-directional 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 interpretants for reaching shared semantics.

	System	head	OnWN	FNWN	SMT	mean
LS	mean	.6552	.6943	.2016	.3005	.5086
	mean TL	.6397	.6808	.1776	.3147	.5028
	min	.6512	.6947	.2003	.2984	.5066
	min TL	.6416	.6853	.1903	.3143	.5055
	max	.6669	.6680	.1867	.2737	.4958
	max TL	.6493	.6805	.1846	.3127	.5059
	$S_1 \rightleftarrows S_2$	.6388	.6695	.1667	.2999	.4938
	$S_1 \rightleftarrows S_2$ TL	.6285	.6686	.0918	.2931	.4816
LPS	mean	.6510	.6971	.1179	.2861	.4961
	mean TL	.6524	.6918	.1940	.3176	.5121
	min	.6608	.6953	.1704	.2922	.5053
	min TL	.6509	.6864	.1792	.3156	.5084
	max	.6588	.6800	.1355	.2868	.4961
	max TL	.6493	.6805	.1846	.3127	.5059
	$S_1 \rightleftarrows S_2$	.6251	.6843	.0677	.2994	.4845
	$S_1 \rightleftarrows S_2$ TL	.6370	.6978	.0951	.2980	.4936
RLPS	mean	.6517	.7136	.1002	.2880	.4996
	mean TL	.6383	.6841	.2434	.3063	.5059
	min	.6615	.7099	.1644	.2877	.5072
	min TL	.6606	.6987	.1972	.3059	.5129
	max	.6589	.7019	.0995	.2935	.5008
	max TL	.6362	.6896	.2044	.3153	.5063
	$S_1 \rightleftarrows S_2$	.6300	.7011	.0817	.2798	.4850
	$S_1 \rightleftarrows S_2$ TL	.6321	.6956	.1995	.3128	.5052

Table 3: Bi-directional STS challenge r and ranking results ranked according to the mean r obtained. We combine the two directions by taking the mean, min, or the max or use the bi-directional RTM model $S_1 \rightleftarrows S_2$.

Acknowledgments

This work is supported in part by SFI (07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University and in part by the European Commission through the QTLaunchPad FP7 project (No: 296347). We also thank the SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational facilities and support.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 435–440, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Ergun Biçici and Josef van Genabith. 2013. CNGL: Grading student answers by acts of translation. In **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics and Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, Georgia, USA, 14-15 June. Association for Computational Linguistics.

Ergun Biçici and Deniz Yuret. 2011a. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283, Edinburgh, Scotland, July. Association for Computational Linguistics.

Ergun Biçici and Deniz Yuret. 2011b. RegMT system for machine translation, system combination, and evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 323–329, Edinburgh, Scotland, July. Association for Computational Linguistics.

Ergun Biçici, Declan Groves, and Josef van Genabith. 2013. Predicting sentence translation quality using extrinsic and language independent features. *Machine Translation*.

Ergun Biçici. 2011. *The Regression Model of Machine Translation*. Ph.D. thesis, Koç University. Supervisor: Deniz Yuret.

Ergun Biçici. 2008. Consensus ontologies in socially interacting multiagent systems. *Journal of Multiagent and Grid Systems*.

Carl Hugo Björnsson. 1968. *Läsbarhet*. Liber.

Chris Bliss. 2012. Comedy is translation, February. http://www.ted.com/talks/chris_bliss_comedy_is_translation.html.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Myroslava O. Dzikovska, Rodney D. Nielsen, and Chris Brew. 2012. Towards effective tutorial feedback for explanation questions: A dataset and baselines. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 200–210, Montréal, Canada, June. Association for Computational Linguistics.

Kenth Hagström. 2012. Swedish readability calculator. https://github.com/keha76/Swedish-Readability-Calculator.

George A. Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, November.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Sameer S. Pradhan, Eduard H. Hovy, Mitchell P. Marcus, Martha Palmer, Lance A. Ramshaw, and Ralph M. Weischedel. 2007. Ontonotes: a unified relational semantic representation. *Int. J. Semantic Computing*, 1(4):405–419.

Yoav Seginer. 2007. *Learning Syntactic Structure*. Ph.D. thesis, Universiteit van Amsterdam.

Alex J. Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wikipedia. 2013. Lix. http://en.wikipedia.org/wiki/LIX.