A Comparison of Greedy and Optimal Assessment of Natural Language Student Input Using Word-to-Word Similarity Metrics

Vasile Rus
Department of Computer Science
The University of Memphis
Memphis, TN 38152
vrus@memphis.edu

Mihai Lintean
Department of Computer Science
The University of Memphis
Memphis, TN 38152
mclinten@memphis.edu

Abstract

We present in this paper a novel, optimal semantic similarity approach based on word-to-word similarity metrics to solve the important task of assessing natural language student input in dialogue-based intelligent tutoring systems. The optimal matching is guaranteed using the sailor assignment problem, also known as the job assignment problem, a well-known combinatorial optimization problem. We compare the optimal matching method with a greedy method as well as with a baseline method on data sets from two intelligent tutoring systems, AutoTutor and iSTART.

Introduction

We address in this paper the important task of assessing natural language student input in dialogue-based tutoring systems where the primary form of interaction is natural language. Students provide their responses to tutor's requests by typing or speaking their responses. Therefore, in dialogue-based tutoring systems understanding students' natural language input becomes a crucial step towards building an accurate student model, i.e. assessing the student's level of understanding, which in turn is important for optimum feedback and scaffolding and ultimately impacts the tutoring's effectiveness at inducing learning gains on the student user.

We adopt a semantic similarity approach to assess students' natural language input in intelligent tutoring systems. The semantic similarity approach to language understanding derives the meaning of a target text, e.g. a student sentence, by comparing it with another text whose meaning is known. If the target text is semantically similar to the known-meaning text then we know the target's meaning as well.

Semantic similarity is one of the two major approaches to language understanding, a central topic in Artificial Intelligence. The alternative approach is full understanding. The full understanding approach is not scalable due to prohibitive costs to encode world and domain knowledge which are needed for full understanding of natural language.

To illustrate the problem of assessing natural language student input in dialogue-based tutoring systems using a semantic similarity approach, we consider the example below from experiments with AutoTutor (Graesser et al., 2005), a dialogue-based tutoring system.

Expert Answer: *The force of the earth's gravity, being vertically down, has no effect on the object's horizontal velocity*

Student Input: *The horizontal component of motion is not affected by vertical forces*

In this example, the student input, also called contribution, is highly similar to the correct expert answer, called expectation, allowing us to conclude that the student contribution is correct. A correct response typically triggers positive feedback from the tutor. The expert answer could also be an

anticipated wrong answer, usually called a misconception. A student contribution similar to a misconception would trigger a misconception correction strategy.

We model the problem of assessing natural language student input in tutoring systems as a paraphrase identification problem (Dolan et al., 2004). The student input assessment problem has been also modeled as a textual entailment task in the past (Rus & Graesser, 2006).

Our novel method to assess a student contribution against an expert-generated answer relies on the compositionality principle and the sailor assignment algorithm that was proposed to solve the assignment problem, a well-known combinatorial optimization problem. The sailor assignment algorithm optimally assigns sailors to ships based on the fitness of the sailors' skills to the ships' needs [7, 8]. In our case, we would like to optimally match words in the student input (the sailors) to words in the expert-generated answer (the ships) based on how well the words in student input (the sailors) fit the words in the expert answer (the ships). The fitness between the words is nothing else but their similarity according to some metric of word similarity. We use the WordNet word-to-word similarity metrics (Pedersen et al., 2004) and Latent Semantic Analysis (Landauer et al., 2007).

The methods proposed so far that rely on the principle of compositionality to compute the semantic similarity of longer texts have been primarily greedy methods (Corley & Mihalcea, 2005; Lintean & Rus, 2012). To the best of our knowledge, nobody proposed an optimal solution based on the principle of compositionality and word-to-word similarity metrics for the student input assessment problem. It is important to note that the optimal method proposed here is generally applicable to compute the similarity of any texts.

We provide experimental results on two datasets provided to us by researchers developing two world-class dialogue-based tutoring systems: AutoTutor (Graesser et al., 2005) and iSTART (McNamara et al., 2004).

Background

It is beyond the scope of this work to offer an exhaustive overview of methods proposed so far to handle the task of assessing natural language student input in intelligent tutoring systems. We only describe next methods that are most relevant to our work.

Assessing student's contributions in dialogue-based tutoring systems has been approached either as a paraphrase identification task (Graesser et al., 2005), i.e. the task was to assess how similar student contributions were to expert-generated answers, or as an entailment task (Rus & Graesser, 2006), in which case the task was to assess whether student contributions were entailed by expert-generated answers. The expert answers were assumed to be true. If a correct expert answer entailed a student contribution then the contribution was deemed to be true as well.

Latent Semantic Analysis (LSA; Landauer et al., 2007) has been used to evaluate student contributions during the dialog between the student by Graesser and colleagues (2005). In LSA the meaning of a word is represented by a reduced-dimensionality vector derived by applying an algebraic method, called Singular Value Decomposition (SVD), to a term-by-document matrix built from a large collection of documents. A typical dimensionality of an LSA vector is 300-500 dimensions. To compute the similarity of two words the cosine of the word's corresponding LSA vector is computed, i.e. the normalized dot-product. A typical extension of LSA-based word similarity to computing the similarity of two sentences (or even larger texts) is to use vector algebra to generate a single vector for each of the sentences (by adding up the individual words' LSA vectors) and then compute the cosine between the resulting sentence vectors. Another approach proposed so far to compute similarities between individual words in the two sentences, greedily selects for each word its best match, and then sums the individual word-to-word similarities in order to compute the overall similarity score for the two sentences (Lintean & Rus, 2012). We do report results with LSA using the latter approach for comparison purposes. Another reason is that only the latter approach allows the application of the optimum matching method.

Extending word-to-word similarity measures to sentence level and beyond has drawn increasing interest in the last decade or so in the Natural Language Processing community. The interest has been driven primarily by the creation of standardized data sets and corresponding shared

task evaluation campaigns (STECs) for the major text-to-text semantic relations of entailment (RTE; Recognizing Textual Entailment corpus by Dagan, Glickman, & Magnini, 2005), paraphrase (MSR; Microsoft Research Paraphrase corpus by Dolan, Quirk, and Brockett, 2004), and more recently for elaboration (ULPC; User Language Paraphrase Challenge by McCarthy & McNamara, 2008).

None of the existing methods for assessing the similarity of texts based on the compositional principle and word-to-word similarity metrics have proposed an optimum method.

Beyond Word-to-Word Similarity Measures

Based on the principle of compositionality, which states that the meaning of longer texts can be composed from the meaning of their individual words (which includes collocations in our case such as "free fall"), we can extend the word-to-word similarity metrics to compute the similarity of longer texts, e.g. of sentences.

In our work, we use a set of WordNet-based similarity metrics as well as LSA. We used the following similarity measures implemented in the WordNet::Similarity package and described in (Pedersen et al., 2004): LCH (Leacock and Chodorow), RES (Resnik), JCN (Jiang and Conrath), LIN (Lin), PATH, and WUP (Wu and Palmer). Some measures, e.g. PATH, are path-based, i.e. use paths of lexico-semantic relations between concepts in WordNet, while some others are gloss-based, that is, they use the text of the gloss or the definition of a concept in WordNet as the source of meaning for the underlying concept.

One challenge with the WordNet word-to-word relatedness measures is that they cannot be directly applied to larger texts such as sentences. They must be extended to larger texts, which we did as described later.

Another challenge with the WordNet word-to-word similarity metrics is the fact that texts express meaning using words and not concepts. To be able to use the word-to-word related measures we must map words in sentences to concepts in WordNet. Thus, we are faced with a word sense disambiguation (WSD) problem. It is beyond the scope of our investigation to fully solve the WSD problem, one of the hardest in the area of Natural Language Processing. Instead, we addressed the issue in two ways: (1) mapped the words in the student contribution and expert answer onto the concepts corresponding to their most frequent sense, which is sense #1 in WordNet, and (2) map the words onto all the concepts corresponding to all the senses and then take the maximum of the relatedness scores for each pair of senses. Because the ALL (all senses) method offered better results and because of space constraints we only report results with the ALL method in this paper.

Greedy versus Optimal Semantic Similarity Matching

This section describes the greedy and optimal matching methods to assess the similarity of two texts based on word-to-word similarity metrics. We assume the two texts, T1 and T2, are two sentences and regard them as bags of words (syntactic information is ignored).

The Greedy Method. In the greedy method, each word in text T1 is paired with every word in text T2 and word-to-word similarity scores are computed according to some metric. The maximum similarity score between words in T1 and any word in T2 is greedily retained regardless of the best matching scores of the other words in T1. The greedily-obtained scores are added up using a simple or weighted sum which can then be normalized in different ways, e.g. by dividing to the longest text or to the average length of the two texts. The formula we used is given in equation 1. As one would notice, this formula is asymmetric, i.e. score(T1,T2) $\neq$ score(T2,T1). The average of the two scores provides a symmetric similarity score, more suitable for a paraphrase task, as shown in Equation 2. In this paper, we do a simple non-weighted sum, i.e. all the words are equally-weighted with a weight of 1.

The obvious drawback of the greedy method is that it does not aim for a global maximum similarity score. The optimal method described next solves this issue.

$$score(T1,T2) = \frac{\sum_{v \in T1} weight(v) * \max_{w \in T2} word-sim(v,w)}{\sum_{v \in T1} weight(v)}$$

Equation 1. *Asymmetric semantic similarity score between texts T1 and T2.*

$$simScore(T1,T2) = \frac{score(T1,T2) + score(T2,T1)}{2}$$

Equation 2. *Symmetric semantic similarity score between texts T1 and T2.*

Optimal Matching. The optimal assignment problem is one of the fundamental combinatorial optimization problems and consists of finding a maximum weight matching in a weighted bipartite graph.

Given a weighted complete bipartite graph $G = X \cup Y; X \times Y)$, where edge xy has weight $w(xy)$, find a matching M from X to Y with maximum weight.

An application is about assigning a group of workers, e.g. sailors, to a set of jobs (on ships) based on the expertise level, measured by $w(xy)$, of each worker at each job. By adding dummy workers or jobs we may assume that X and Y have the same size, *n*, and can viewed as $X = \{x_1, x_2, \ldots, x_n\}$ and Y = $\{y_1, y_2, \ldots, y_n\}$. In the semantic similarity case, the workers and jobs are words from the two sentences to be compared and the weight $w(xy)$ is the word-to-word similarity between word x and y in the two sentences, respectively.

The assignment problem can be stated as finding a permutation π of {1, 2, 3, … , n} for which $\sum_{i=1}^{n} w(x_i y_{\pi(i)})$ is maximum. Such an assignment is called optimum assignment. An algorithm, the Kuhn-Munkres method (Kuhn, 1955), has been proposed that can find a solution to the optimum assignment problem in polynomial time. For space reasons, we do not show here the algorithm in detail.

To illustrate the difference between the two methods, we use the two sentence fragments shown in Figure 1. A greedy method would pair *motion* with *motion* (similarity score of 1.00) as that is the maximum similarity between *motion* and any word in the opposite sentence and *acceleration* is paired with *speed* (similarity score of 0.69) for a total score of 1.69 (before normalization). An optimal matching would yield an overall score of 1.70 by pairing *motion* in the first sentence with *speed* (similarity of 0.75) and *acceleration* with *motion* (similarity of 0.95).

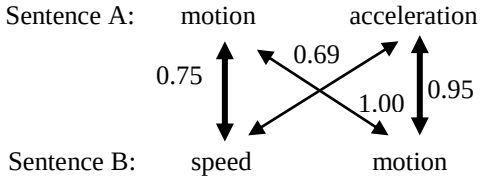

Figure 1. *Examples of two sentence fragments and word-to-word similarity scores for each of the word pairs across sentences. The bold arrows show optimal pairing.*

Experimental Setup and Results

We present in this section the datasets we used in our experiments and the results obtained. As we already mentioned, we use two datasets containing real student answers from two dialogue-based tutoring systems: AutoTutor (Graesser et al., 2005) and iSTART (McNamara et al., 2004).

The AutoTutor dataset contains 125 student contribution – expert answer pairs and the correct paraphrase judgment, TRUE or FALSE, as assigned by human experts. The target domain is conceptual physics. One expert physicist rated the degree to which particular speech acts expressed during AutoTutor training matched particular expert answers. These judgments were made on a sample of 25 physics expectations (i.e., correct expert answers) and 5 randomly sampled student answers per expectation, yielding a total of 125 pairs of expressions. The learner answers were always responses to the first hint for that expectation. The E-S pairs were graded by Physics experts on a scale of 1-4 (4 being perfect answer). This rubric could be mapped onto a binary TRUE-FALSE rubric: scores 3 and 4 equal a TRUE decision and 1 and 2 equal a FALSE decision. We ended up with 36 FALSE and 89 TRUE entailment pairs, i.e. a 28.8% versus 71.2% split (as compared to the 50-50% split of RTE data).

The iSTART data set, also known as the User Language Paraphrase Corpus (McCarty & McNamara, 2008) comprises annotations of paraphrase relations between student responses and ideal answers. The corpus contains 1998 pairs collected from previous student iSTART sessions and is divided into training (1499 instances) and testing (499 instances) subsets. The training subset contains 54% positive instances while testing contains 55% positive instances. The iSTART texts represent high school students' attempts to self-explain biology textbook texts.

To evaluate the performance of our methods, we compare the methods' judgments with the expert judgments. The percentage of matching judgments provides the accuracy of the run, i.e. the fraction of correct responses. We also report kappa statistics which indicate agreement between our methods' output and the human-expert judgments for each

instance while taking into account chance agreement.

Tables 1, 2, and 3 summarize the results on the original AutoTutor data (from Rus & Graesser, 2006; Table 1), the re-annotated AutoTutor data by a second rater with inter-annotator agreement of 0.606 (Table 2), and the ULPC test subset (Table 3). For the ULPC corpus the methods have been trained on the training subset, an optimum threshold has been learned (such that scores above the threshold mean TRUE paraphrases) which is then used on the test data. Since the AutoTutor dataset is small, we only report results on it as a whole, i.e. only training. We report for each corpus a baseline method of guessing all the time the dominant class in the dataset (which is TRUE paraphrase for all three datasets), a pure greedy method (Greedy label in the first column of the tables), a greedy method applied to the words paired by the optimum method (optGreedy), and the results with the optimum matching method (Optimum).

Overall, the optimum method offered better performance in terms of accuracy and kappa statistics. The greedy method yields results that are close. In fact, when analyzed as raw scores instead of binary decisions (as is the case when computing accuracy) the greedy raw score are on average very similar to the optimum scores. For instance, for the LSA word-to-word similarity metric which provided best accuracy results on the ULPC dataset (accuracy=.643 for optimum and .615 for greedy), the average raw scores are .563 (using optimum matching) and .567 (using greedy matching). One reason for why they are so closed is that in optimum matching we have one-to-one word matches while in the greedy matching many-to-one matches are possible. That is, two words v and w from text T1 can be matched to same word y in text T2 in the greedy method. If we enforce that only one-to-one matches are possible in the greed method as in the optimum method, then we obtain the optGreedy method. The optGreedy method does work better than the pure greedy method (Greedy in the tables).

Another reason for why the raw scores are close for greedy and optimum is the fact that student input and expert answers in both the AutoTutor and ULPC corpora are sharing many words in common (>.50). This is the case because the dialogue is highly contextualized around a given, e.g. physics, problem. In the answer, both students and experts refer to the entities and interactions in the problem statement which leads to high identical word overlap. Identical words lead to perfect word-to-word similarity scores (=1.00) increasing the overall similarity score of the two sentences in both the greedy and optimum method.

Conclusions and Future Work

Overall, the optimum method offers better performance in terms of accuracy and kappa statistics than greedy and baseline methods.

The way we modeled the student assessment problem in this paper cannot deal with some type of responses. For instance, sometimes students' responses are mixed. Instead of being TRUE or FALSE responses, they contain both a correct part and an incorrect part as illustrated in the example below (Expert Answer provided for reference).

Expert Answer: The object continues to have a constant horizontal velocity component after it is released that is the same as the person horizontal velocity at the time of dropping the object.

Student Input: *The horizontal velocity will decrease while the vertical velocity increases.*

Such a mixed student input should trigger a mixed feedback from the system: "You are partially right! The vertical velocity will increase but not the horizontal velocity. Can you explain why?" We plan to address this problem in the future by proposing a more sophisticated model.

We also plan to answer the question of how much lexical versus world and domain knowledge each of these measures can capture. For instance, WordNet can be viewed as capturing some world knowledge as the concepts' definitions provide information about the world. However, it might be less rich in capturing domain specific knowledge. Indeed, WordNet seems to capture less domain knowledge at first sight. For instance, the definition of *acceleration* in WordNet does not link it to the concept of *force* but physics laws do, e.g. Newton's second law of motion.

Acknowledgments

This research was supported in part by the Institute for Education Sciences (award R305A100875). The opinions and findings reported in this paper are solely the authors'.

ID	RES	*LCH*	*JCN*	*LSA*	*Path*	*Lin*	*WUP*
Baseline	*.712*	*.712*	*.712*	*.712*	*.712*	*.712*	*.712*
Greedy	.736/.153	*.752/.204*	.760/.298	.744/.365	.752/.221	.744/.354	.760/.298
optGreedy	.744/187	*.752/.221*	.760/.298	.744/.306	.752/.309	.752/.204	.784/.349
Optimal	.744/.236	*.752/.204*	.760/.298	.744/.221	.752/.334	.752/.204	.784*/.409*

Table 1. Accuracy/kappa on AutoTutor data (* indicates statistical significance over the baseline method at p<0.005 level).

ID	RES	*LCH*	*JCN*	*LSA*	*Path*	*Lin*	*WUP*
Baseline	*.568*	*.568*	*.568*	*.568*	*.568*	*.568*	*.568*
Greedy	.616/.137	*.608/.117*	.624/.214	.632/.256	.624/.161	.608/1.34	.624/.181
optGreedy	.632/.192	*.632/.207*	.632/.229	.624/.218	.632*/.177*	.624/.165	.648*/.235*
Optimal	.624*/.153*	*.624/.169*	.640*/.208*	.640/.283	.624/.165	.624*/.148	.624/.173

Table 2. Accuracy/kappa on AutoTutor data with user annotations (* indicates statistical significance over the baseline method at p<0.005 level).

ID	RES	*LCH*	*JCN*	*LSA*	*Path*	*Lin*	*WUP*
Baseline	*.547*	*.547*	*.547*	*.547*	*.547*	*.547*	*.547*
Greedy	.619/.196	*.619/.201*	.629/.208	.615/.183	.635/.221	.629/.214	.621/.201
optGreedy	.621/.195	*.615/.201*	.629/.208	.643/.237	.623/.197	.619/.196	.613/.190
Optimal	.625/.205	*.615/.196*	.629/.208	.643/.237	.633/.215	.623/.203	.625/.214

Table 3. Accuracy/kappa on ULPC test data (all results are statistically different from the baseline at p<0.005 level).

References

Courtney Corley and Rada Mihalcea. 2005. Measures of Text Semantic Similarity. In Proceedings of the ACL workshop on Empirical Modeling of Semantic Equivalence, Ann Arbor, MI, June 2005.

Ido Dagan, Oren Glickman, and Bernando Magnini. 2005. The PASCAL recognizing textual entailment challenge. In Proceedings of the PASCAL Workshop.

Bill W. Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In Proceedings of the 20th International Conference on Computational Linguistics, Geneva, Switzerland.

Arthur C. Graesser, Andrew Olney, Brian C. Hayes, and Patrick Chipman. 2005. Autotutor: A cognitive system that simulates a tutor that facilitates learning through mixed-initiative dialogue. In Cognitive Systems: Human Cognitive Models in System Design. Mahwah: Erlbaum.

Harold W. Kuhn. 1955. "The Hungarian Method for the assignment problem", Naval Research Logistics Quarterly, 2:83–97, 1955. Kuhn's original publication.

Thomas K. Landauer, Danielle S. McNamara, Simon Dennis, and Walter Kintsch. 2007. Handbook of Latent Semantic Analysis. Mahwah, NJ: Erlbaum.

Mihai Lintean and Vasile Rus. 2012. Measuring Semantic Similarity in Short Texts through Greedy Pairing and Word Semantics. To be presented at The Twenty-Fifth International FLAIRS Conference. Marco Island, Florida.

Philip M. McCarty and Danielle S. McNamara. 2008. User-Language Paraphrase Corpus Challenge, online.

Danielle S. McNamara, Irwin B. Levinstein, and Chutima Boonthum. 2004. iSTART: interactive strategy training for active reading and thinking. Behavioral Research Methods, Instruments, and Computers, 36(2).

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity – Measuring the Relatedness of Concepts. In Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2004).

Vasile Rus, and Arthur C. Graesser. 2006. Deeper Natural Language Processing for Evaluating Student Answers in Intelligent Tutoring Systems, Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06).