Proceedings of the ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2015
August 2-5, 2015, Boston, Massachusetts, USA

DETC2015-47358

LARGE SCALE NEEDS-BASED OPEN INNOVATION VIA AUTOMATED SEMANTIC TEXTUAL SIMILARITY ANALYSIS

Cory R. Schaffhausen
Department of Mechanical Engineering
University of Minnesota
Minneapolis, MN 55455
Email: schaf390@umn.edu

Timothy M. Kowalewski, PhD *
Department of Mechanical Engineering
University of Minnesota
Minneapolis, MN 55455
Email: timk@umn.edu

ABSTRACT

Open innovation often enjoys large quantities of submitted content. Yet the need to effectively process such large quantities of content impede the widespread use of open innovation in practice. This article presents an exploration of needs-based open innovation using state-of-the art natural language processing (NLP) algorithms to address existing limitations of exploiting large amounts of incoming data. The Semantic Textual Similarity (STS) algorithms were specifically developed to compare sentence-length text passages and were used to rate the semantic similarity of pairs of text sentences submitted by users of a custom open innovation platform. A total of 341 unique users submitted 1,735 textual problem statements or unmet needs relating to multiple topics: cooking, cleaning, and travel. Scores of equivalence generated by a consensus of ten human evaluators for a subset of the needs provided a benchmark for similarity comparison. The semantic analysis allowed for rapid (1 day per topic), automated screening of redundancy to facilitate identification of quality submissions. In addition, a series of permutation analyses provided critical crowd characteristics for the rates of redundant entries as crowd size increases. The results identify top modern STS algorithms for needfinding. These predicted similarity with Pearson correlations of up to .85 when trained using need-based training data and up to .83 when trained using generalized data. Rates of duplication varied with crowd size and may be approximately linear or appear asymptotic depending on the degree of similarity used as a cutoff. Semantic algorithm performance has shown rapid improvements in recent years. Potential applications to screen duplicates and also to screen highly unique sentences for rapid exploration of a space are discussed.

1 INTRODUCTION

Open innovation is a method for seeking ideas for innovation from external sources. The importance of this trend has been previously discussed and shown to be successful in several applications [1–4]. However, open innovation processes result in large quantities of submitted content [5, 6], and the resources required to assess and filter redundancy and quality impede the use of open innovation in practice. Survey results of companies with open-innovation experience note complaints that reviewing and assessing externally-submitted ideas takes "an army of internal people" [7, p. 15]. A recent Cisco open innovation project required a team of six full-time employees working for three months in order to evaluate 1200 submissions [6].

Commercial idea management systems exist such as Ideascale (www.http://ideascale.com) and The IdeaWall (www.theideawall.com). Ideascale includes keyword search for predictive duplication merging. While exact methods are not listed for these commercial systems, existing keyword methods such as Lucene [8] have previously been used. However, existing systems must still rely on administrator oversight or knowledge of users to identify and flag duplicates. Relationships between ideas may be more complex than only duplication; however, annotations of relationship hierarchies also remain

*Address all correspondence to this author.

largely manual [9]. Idea overflow and redundancy remain a challenge for managing open innovation data [5, 9]. Assessing similarity using pairwise comparisons of submissions analyzed with automated NLP algorithms may identify redundancy and reduce resources needed to manage data.

This study makes important contributions: (1) confirming automated algorithms can potentially overcome the resource-intensive process of assessing open innovation data, (2) presenting recommended algorithms based on rigorous comparisons, and (3) demonstrating expected rates of duplication for needs statements submitted from large crowds for the topics used.

1.1 Natural Language Processing Background

NLP algorithms utilizing machine learning are increasingly studied and are rapidly improving [10,11]. A goal of natural language understanding—a subtopic of NLP—is to comprehend the intended semantic content of text. This is of particular relevance to textual design processing such as needfinding or analyzing textual innovation content. Modern techniques have moved well beyond keyword analysis or parts-of-speech comparison to extracting the concepts in or semantic meaning of sentences, phrases, and passages. The increasing trend towards employing probabilistic machine learning techniques ensure that semantic content can be automatically extracted from otherwise prohibitively large corpora, for example, from phrases never seen in the original training set used to tune the algorithms. While still in its infancy, it is evident that the accuracy and speed of these semantic techniques continue to improve with increasing momentum [11]. Simultaneously, the need for arduous supervision and human input during training or use continues to decrease [11]. This suggests that certain NLP approaches can not only enable the automated semantic processing of textual design content but do so at large scales (e.g. large scale needfinding).

Semantic Textual Similarity (STS) is a form of NLP analysis used to measure the similarity of two phrases or sentences. Distinguishing elements of STS are graded and symmetric ratings. A graded rating refers to a sentence pair being "more" or "less" similar than another sentence pair. A symmetric rating indicates there is no directionality when comparing sentence A to sentence B (e.g. A to B is the same as B to A). STS also provides a framework to cohesively combine a number of different NLP components, such as word sense disambiguation and induction, lexical substitution, and semantic role labeling, into a single evaluation [12]. Because of these unique characteristics, STS may be a useful tool to perform pairwise comparisons and assess redundancy in open innovation content.

Special cases of open innovation have been used to generate large numbers of user-generated product needs, rather than ideas [13]. While open innovation has historically not been used for the direct solicitation of user needs, Faste has suggested that advances in online knowledge management "could be applied to crowdsourced needfinding research" [14, p. 5], and further states "Perhaps one of the most important ways in which open-innovation can therefore be made to thrive is by enabling individuals to report their own needs." [14, p. 4]. In addition to obvious applications in idea management, STS methods are appropriate for novel open innovation tasks such as evaluating large sets of user-submitted needs. Needs descriptions are inherently text-based and might represent complex, nuanced aspects of a problem, thus necessitating state-of-the-art analysis methods.

This study selected two STS algorithms based on performance and ease of use. Both had previously been a top-performer at international NLP competitions, allowed Python script querying, and were freely available. Algorithms were trained and tested using multiple data sets and then used to rate similarity for need-based open innovation.

1.2 SemEval Algorithm Competition Background

SemEval is a series of evaluations coinciding with the *SEM conference (Joint Conference on Lexical and Computational Semantics). For each conference, the organizers supply standard annotated data sets in a wide variety of NLP tasks, including English and multilingual versions. As is typical in machine learning research, an algorithm is "tuned" to an application area by setting internal parameters during a training step that invokes a training data set. Next, the algorithm's performance is evaluated on a different data set, not included in the training set, which is referred to as the evaluation set or test set.

During SemEval, research teams submit algorithms to run the supplied training and test data, and the outcomes are ranked based on evaluation metrics, such as Pearson correlation to gold standard data (human ratings). The 2012 SemEval was the first to introduce a semantic textual similarity (STS) rating task, and a similar task was repeated in 2013. Each year had over 30 participating teams. In this task, the similarity of two text passages (e.g. sentences) is computed on a scale of 0 (different topics) to 5 (completely equivalent) [12, 15].

The STS task provided several different data sets. The 2012 task used training and test data derived from the same sets. The 2013 task used the same training data as 2012 but provided several new data sets for testing. One example of a provided data set is the MSR Video Paraphrase Corpus (MSRvid) set, originating from Microsoft Research. The data was collected from human participants who were describing a short video clip. These descriptions were combined with descriptions of other similar and different video clips to produce a set of 1500 sentence pairs with a wide range of similarity [12].

1.3 Algorithm 1 Background: TakeLab-simple

The TakeLab-simple system was one of two 2012 SemEval submission from the TakeLab research group (University of Zagreb, Croatia) for the 2012 STS task. The final mean ranking

of the system was 2nd overall for 2012. The TakeLab group provided open-source files for TakeLab-simple following the conference. The TakeLab-simple algorithm combines a variety of tools into an aggregate similarity score for two text passages. These tools included knowledge-based word similarity using WordNet, corpus-based word similarity using Latent Semantic Analysis (LSA), and several others [12, 16]

1.4 Algorithm 2 Background: UMBC-PairingWords

The UMBC-PairingWords system was one of the three 2013 SemEval submissions from the UMBC Ebiquity research group (University of Maryland, Baltimore County and Johns Hopkins University). The final mean ranking of the system was 1st overall for 2013 in the CORE task. The UMBC group provides an online interface and Python-based code to query the existing system (http://swoogle.umbc.edu/SimService/index.html). The UMBC algorithm also combines a variety of tools for an aggregate similarity score [15, 17]

2 METHODS

This study analyzed data previously collected during a novel application of open innovation. This open innovation scenario tested the ability of users to view a variety of priming and stimulus types and to then directly articulate user needs, independent of proposed or embedded solutions [13]. The two STS algorithms described in Sections 1.3-1.4, tested similarity of data sets comprised of descriptions of user needs with the objective of differentiating between sets of duplicate and unique need statements. A summary of the multiple training, test, and analysis data sets is shown in Figure 1.

2.1 Need Statement Data Description

A preliminary data collection study collected the need statement analysis set (IRB exempt, Study No. 1309E42581). The objective was to collect as many need statements as possible, and the instructions were intentionally framed to motivate a focus on quantity and to create a process comparable to brainstorming ideas.

The Amazon Mechanical Turk (AMT) worker site (https://www.mturk.com/) was used for recruiting participants. AMT is a system allowing requestors to submit small, fee-for-service tasks. Tasks are referred to as Human Intelligence Tasks (HITs). The pay for individual HITs is typically proportional to the task duration, and the approximate range is $.10 per minute.

The participants recruited for the user needs collection were directed to a custom survey interface developed using Zoho Creator (Zoho Corp., Pleasanton, CA). Figure 2 shows the user interface available for entering need statements and stories. Participants were required to have an approval rate of 95% or higher

FIGURE 2. User Interface for Entering Needs

for completed work, a history of at least 100 completed HITs, and a United States IP address location.

Participants submitted needs related only to a single, randomly assigned topic, selected from three options: preparing food and cooking, doing housecleaning and household chores, and planning a trip. All participants reviewed consent information and continued to a training exercise including basic instructions and a quiz. The instructions stated that inventions should not be included and problems should be described in a complete sentence. Participants then took a quiz related to a new topic (reading books) and selected which sample statements followed the instructions. The training was paid as a fixed amount of $0.65 for both pass and fail outcomes, and those who failed were not able to continue.

The study interface allowed the participant to submit open-ended need statements. This included a single-sentence description of a need and an optional paragraph-length description of background information and context. The participants were able to enter as many statements as possible, and were compensated per entry at a rate of $.05 per need statement and $.15 per optional story (using the AMT bonus system).

The interface to submit needs was combined with a method to display stimulus information to potentially generate additional needs that were not obvious or easy to articulate when unaided. The three stimulus types included viewing one of the following: a narrative prompt, a group of previously-submitted need state-

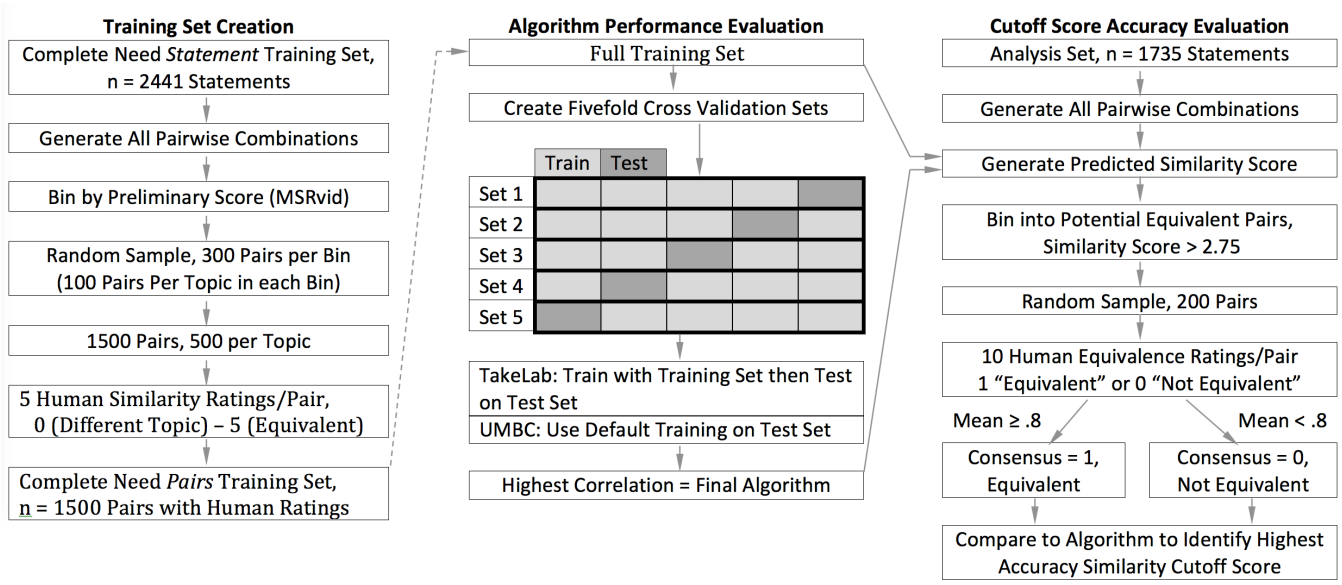

FIGURE 1. Overview of Study Work Flow, Data, and Analyses

ments (by other participants), and a group of images related to the topic. Participants could choose any stimulus type, and there was no limit to the number of selections.

The narrative prompt stimulus asked participants to try to think about a particular type of need, often relating to a specific type of experience or emotional state (e.g. a recent frustrating experience). The stimulus need statements were collected during previous similar studies, and each need statement was paired with the optional background story. The stimulus images were collected from AMT workers during a previous pilot study. Each stimulus type consisted of 30 or more potential content pages, which were randomly selected and were not duplicated per participant.

Table 1 provides an example of each of the types of stimulus information. Stimulus information was displayed adjacent to the need entry and could be viewed simultaneously. The detailed methods and effects of stimulus type and incentives has been previously described [13].

This method was used to generate the analysis set of need statements for automated similarity ratings. The analysis set is summarized in Table 2. Algorithms were trained using a training set of need statements independently collected during early, validation studies of the open innovation method.

2.2 Need Statement Preprocessing

The list of submitted needs consisted of a sequential Need ID number, a need statement, an optional story, an AMT user ID, a record of all stimulus information previously viewed, and a topic area key. Need statements were separated based on topic to create three independent data sets.

Preprocessing steps were executed using Python scripts. A first step removed contractions and non-standard characters. A second step generated tab-separated files of full-text sentence pairs as input files for the algorithm similarity rating. These input files provided necessary combinations in order to compare each need to all others. The results would determine which statements might be duplicate or redundant. A matrix with all needs on both the horizontal and vertical axes provided all pairwise combinations. A need statement would be compared to itself; however, if two statements, A and B, were represented with A:B, then the reverse B:A would be redundant and was not required (half of the complete matrix was used).

2.3 Need Statement Training Sets

The complete need statement training set (see Figure 1) was processed into pairs and initially rated using a provided, trained algorithm. This algorithm was the TakeLab system trained using provided MSRvid training data. The result was a preliminary score to help create a smaller subset with an approximately uniform distribution of similar and unique pairs. The ratings were binned by preliminary score, and randomly sampled for 300 pairs per bin. The complete need pairs training set was comprised of 1500 pairs (500 pairs per topic) across the 0-5 score range. This training set was rated using AMT workers in a fashion similar to MSRvid data [12]. In order to evaluate algorithm performance, the complete need pairs training set was partitioned into five training and five test sets (each with 100 pairs per topic) for a fivefold cross validation.

TABLE 1. Examples of each stimulus type

Example narrative prompt for cooking	Example need and story for cleaning	Example image for planning a trip
"Think of a time when you were preparing food and cooking, and you were unsure how to get the result you wanted. Maybe this was new for you or the things you were trying werent working the way you expected. You didnt know the best way to move ahead or didnt know where to find the information you needed. What problem could be addressed to help you understand what to do?"	Need: "I wish there was an easier way to fold fitted sheets." Story: *"Impossible. It's just impossible. I've tried viewing tutorials on YouTube and following written instructions, but I can't do it. Everything else in the closet is neatly folded, except for the fitted sheet which is just a big wad of wrinkled linen."*	

2.4 Similarity Cutoff Scores

The analysis of potential duplicates used a cutoff score to divide pairs of need statements into potential duplicates or potential unique entries. The analysis assumed the cutoff score would represent a point where two statement were considered equivalent in meaning. In order to evaluate the accuracy of a cutoff score, a sample of need statements was taken from the need pair analysis set with predicted similarity ratings of 2.75-5 (see Figure 1). This set of 200 test pairs was in the approximate range of an equivalency cutoff. These pairs were rated using AMT with 10 ratings per pair. The rating was a binary selection of 1 ("Equivalent") or 0 ("Not Equivalent"). The mean equivalence rating of each pair was calculated. A mean value of .8 or higher was used to represent consensus that the pair was equivalent, and a consensus value was set to 1. Mean values of less than .8 represented not equivalent, and a consensus value was set to 0.

The predicted similarity scores using TakeLab-simple trained with need statement data was compared to the binary consensus values (0 or 1) for the 200 test pairs. The accuracy of predicted ratings for a range of cutoff scores was plotted using the ROCR package in R [18], and a cutoff representing equivalent meaning was selected based on best-case accuracy.

Additional cutoff scores were evaluated during later analyses to test the effect of filtering data based on criteria other than equivalency (e.g. mostly similar or somewhat similar).

2.5 Algorithm 1 Analysis: TakeLab-simple

The TakeLab source files included word corpus files (New York Times Annotated Corpus and Wikipedia Corpus) that had been previously filtered to include only the words present in the SemEval data sets. The complete corpus files were obtained and filtered for only the words present in the set of all training and analysis need statements. This step creates manageable file sizes and does not impact ratings.

The TakeLab-simple system was trained on each of the five need statement validation training sets. The corresponding five validation test sets were analyzed, producing an output value of the predicted similarity score on a scale of 0 to 5. A Pearson correlation value was used to compare predicted similarity scores to human ratings. Correlation values were calculated for the five test sets using both the original SemEval MSRvid model and each respective new model from need statement training data.

Following the cross validation, the TakeLab-simple system was trained using the complete training set of 1500 pairs. This final need statement training model was used for the entire need statement analysis set as described in Section 2.1.

2.6 Algorithm 2 Analysis: UMBC-PairingWords

The UMBC system was not tested as a local system; therefore, the system was trained using only (default) SemEval data. The Python API accesses the system via a URL consisting of embedded pairs of text passages. The output value is a predicted similarity score in the range of 0 to 1, and this output is scaled to match the SemEval range of 0 to 5. The UMBC-PairingWords system has an additional parameter to select one of 3 configuration types, denoted as type 0, 1, or 2. This value modifies the number of parameter combinations tested (1, 2, and 4, respectively) to determine maximum similarity. The combinations include parameters such as "ignore common adverbs", "use extended stopwords" and "support acronym" (Lushan Han, UMBC, personal communication).

The 5 test sets from the cross validation study were analyzed accessing the system via embedded URL parameters. All three configuration types were tested. A Pearson correlation value was used to compare predicted similarity scores to human ratings.

2.7 Identifying Unique and Redundant Entries

The algorithm performance evaluation and the cutoff score evaluation (see Figure 1) determined the final algorithm and cutoff score used to test redundant entries. The need statement analysis set was processed as described in Section 2.2. Resulting pairs were analyzed using the Takelab-simple system trained using needs data as described in Section 2.3. The final analysis used only the system with the highest Pearson correlation values as reported in Section 3.

The set of potential duplicate sentence pairs was assumed to be those with a similarity score above the cutoff as described in Section 2.4. Although the complete analysis of similarity included pairs where a single sentence was compared to itself, these pairs were omitted from the set of potential duplicates as these pairs did not represent a sentence submitted multiple times.

If a single sentence was included in the set of potential duplicates multiple times, the total count of pairs was recorded for each baseline sentence. Within a pair of sentences, the baseline was considered to be the sentence submitted first (e.g. with the lowest ID number).

The accuracy of predicted ratings above and below the cutoff score was evaluated using human ratings of equivalency as described in Section 2.4. A false negative rating would be a high human consensus of equivalence, but a low predicted similarity score. A false positive would be a low human consensus of equivalence, but a high predicted similarity score.

2.8 Analysis of Crowd Size Permutations

The analysis described in Section 2.7 includes needs submitted from all users in each topic group. A random permutation analysis was used to determine how the relative quantity of unique needs changes with increasing group size. Figure 3 shows a schematic representation of analyzing one permutation. Analyzing group size characteristics required input files with each sentence replaced with the sequential need ID. This was appended with the user ID of the participant submitting each need and the similarity score of the sentence pair.

The total list of users for each group was randomly sampled with sizes of 1, 5, 10,... n, where n equals the total users for each topic. Each group sample size was repeated for 50 different permutations. For each group permutation, the complete list of sentence pairs was filtered to only include pairs where both sentences were submitted by users in the permutation group. Pairs where a sentence was compared to itself (same ID) were omitted. This complete list was divided into potential duplicate pairs (score $\geq$ cutoff) and potential unique pairs (score $<$ cutoff).

The list of potential unique pairs was then compared to the list of potential duplicates and if a sentence ID was found in the potential unique list that was present in the potential duplicate list, the duplicate sentence ID was replaced with the baseline ID of the duplicate pair. As before, the baseline was the first sentence to be submitted out of any pair of potential duplicates.

After substituting for all potential duplicates, each resulting list of unique sentence pairs was assessed for the count of all unique sentence ID's. This value was calculated for all 50 permutations of a given group size, and the mean and standard error for each group size was calculated and plotted.

In order to determine the effects of cutoff scores representing varying degrees of similarity, the permutation analysis was repeated for a range of cutoff scores from 1 to 4.

TABLE 2. Summary of Need Statement Data Collection

Topic	Users	Need statements	Combination pairs
Cooking	104	568	161,596
Cleaning	121	650	211,575
Travel	116	517	133,903
Total	**341**	**1,735**	**507,074**

TABLE 3. Algorithm Performance Compared to Human Ratings

	Manually Trained		"Off-the-Shelf"		
Test Set	TakeLab (MSRvid)	TakeLab (Needs)	UMBC (type 0)	UMBC (type 1)	UMBC (type 2)
Set 1	.70	.85	.66	.66	.84
Set 2	.72	.89	.63	.63	.85
Set 3	.71	.84	.64	.64	.83
Set 4	.71	.86	.63	.63	.83
Set 5	.67	.82	.83	.83	.82
Mean	.70	**.85**	.68	.68	**.83**

3 RESULTS

The data collection process generated 1,735 need statements for the analysis set. After dividing into 3 topics and generating pairwise combinations, the total of three sets of sentence pairs was 507,074. Table 2 shows a summary of the counts of need statements and pairwise combinations for each topic group.

3.1 Performance of Algorithms

Table 3 lists Pearson correlation values for all STS systems. The TakeLab-simple system was tested for models trained using MSRvid data and each cross validation training set. The UMBC-PairingWords was tested using three configuration settings. The UMBC-PairingWords system was available only as trained with SemEval data. All comparisons were relative to human ratings of the test sets as described in Section 2.3.

The TakeLab-simple system trained using a need statement training set resulted in a mean Pearson correlation of .85 and the UMBC system trained using SemEval data resulted in a value of .83 for the type 2 configuration setting.

3.2 Summary of Potential Duplicates

Table 4 provides the number of sentences in each topic with a potential duplicate based on a cutoff score of 4.0. Table 5 includes example text of potential duplicate sentences with the highest similarity scores in each topic. Similarity scores relative to baseline sentences are provided for each. Text shown is after

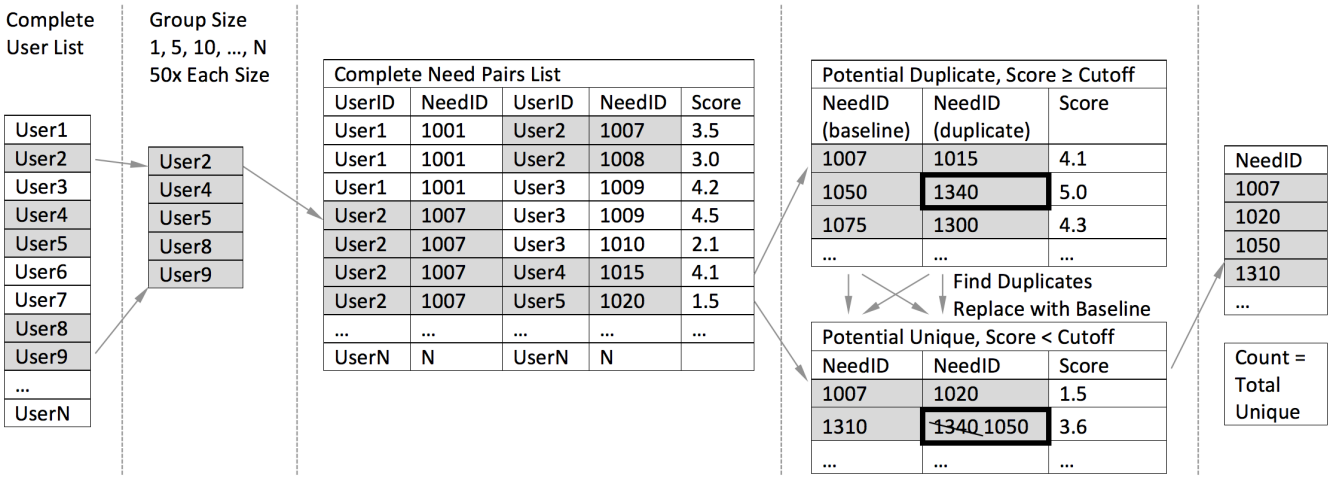

FIGURE 3. Schematic of Group Size Permutation Analysis

TABLE 4. Summary of Potential Duplicates at Cutoff = 4

Topic	Total Pairs, Score ≥ 4	Max. Duplicates per Sentence
Cooking	7	1
Cleaning	21	6
Travel	6	1

preprocessing as described in Section 2.2.

3.3 False Negatives and False Positives

The accuracy of predicted ratings over a range of cutoff scores is shown in Figure 4. The average accuracy plateaus at approximately .75 at a cutoff value of 4.0. A cutoff value of 4.0 was therefore used to divide predicted values into lists of potential duplicates and potential unique statements. One potential source of inaccuracy is the demonstrated variability, or lack of consensus, in human ratings of equivalence. Figure 5 shows a wide band of pairs with an equivalence rating between .3 and .7. This represents the range where at least 3 out of 10 raters differed from the majority.

Tables 6 and 7 show worst-case examples of inaccurate predicted ratings. A single pair from each topic was selected based on the highest discrepancy between predicted similarity scores and human-rated equivalence.

3.4 Unique Statements and Crowd Size

Figure 6 shows plots of each topic for the mean quantity of unique need statements at each group size. Points represent the mean count of unique needs for 50 permutations of the group

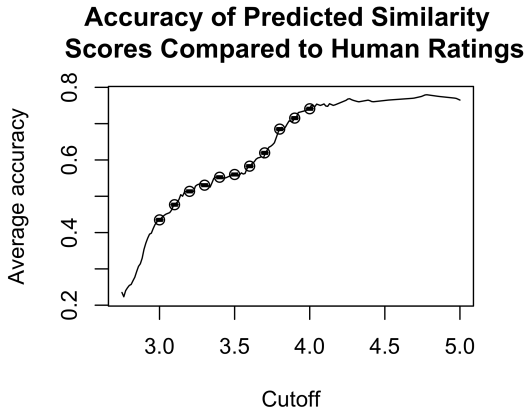

FIGURE 4. Accuracy of Predictions, Points Shown at Values for Cutoff = 3 - 4, by .1

size. Curves for unique needs vs. group size are repeated for a range of cutoff values representing very little similarity (cutoff = 1) to equivalent (cutoff = 4). Each plot demonstrates that the count of unique needs increases nearly linearly at high cutoff values, as there are very few pairs above this cutoff and very few substitutions are made due to duplicates. As the cutoff value decreases, the number of unique needs appears asymptotic for large crowd sizes. Figure 7 shows all topics plotted together using a cutoff score of 4, representing equivalent meaning of sentence pairs. While the slopes of each curve for the 3 topics are similar, at a group size of 100, the count of needs in the travel group is approximately 80 lower than the count in Cleaning and Cooking groups. In Figs. 6 and 7, error bars are shown but are smaller

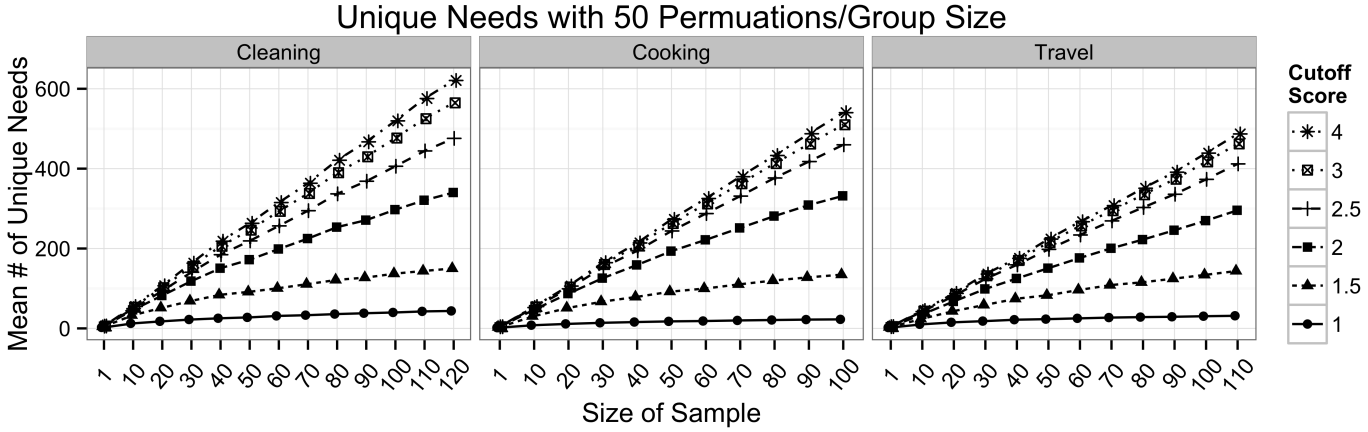

FIGURE 6. Quantities of Unique Statement with Increasing Group Sizes [Standard Error (SE) Bars Smaller Than Points]

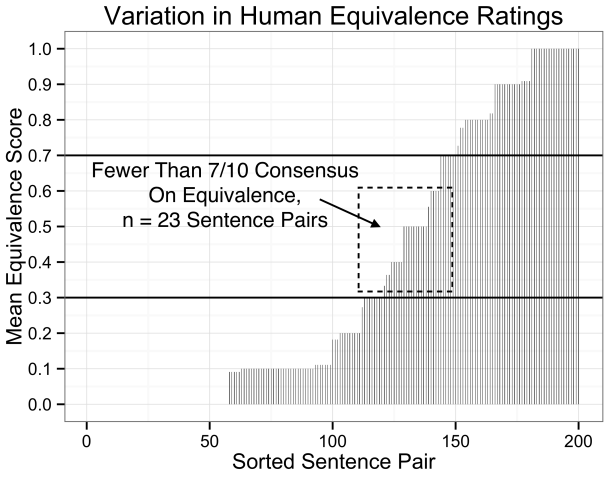

FIGURE 5. Sorted Mean Equivalence Scores

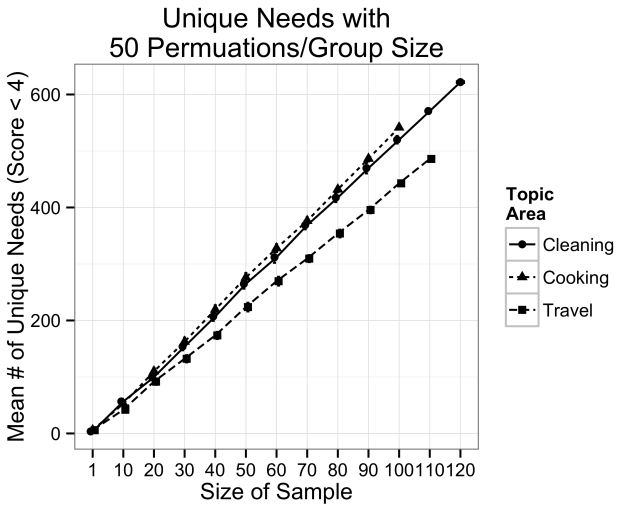

FIGURE 7. Quantities of Unique Statements at Cutoff Score = 4 [Standard Error (SE) Bars Smaller Than Points]

than displayed data points.

4 DISCUSSION

This study confirms NLP algorithms can potentially overcome the resource-intensive process of assessing open innovation data through automated screening of duplicates. Two state-of-the-art algorithms are presented and offer the ability to generate accurate predictions using both task-specific training data and generalized training data.

4.1 STS Can Detect Duplications and Uniqueness in Needs-Based Open Innovation Data

The results show the ability of STS algorithms to detect duplicate need statements among three independent lists of more than 500 sentences each. Table 5 shows examples of both exact duplicates as well as statements using similar language to convey equivalent meaning. Correlations were generally very high (up to .85 when using an algorithm trained with need-statement data). The results further support future work to apply STS methods to needs-based open innovation data as well as commonly used applications for potential solutions and ideas if submitted via text.

TABLE 5. Highest Similarity Sentences and Scores

Score	Need Statements: Cooking, Cleaning, Travel (top to bottom)
Baseline	I need a better way to store lids for my pots and pans.
5.0	I need a way to store my pots and pans.
Baseline	I need a way to keep cool while cooking in the kitchen.
5.0	A way to keep cool in the kitchen while cooking
Baseline	A way to make food cook more evenly in the microwave
4.97	I wish there were a way to make food cook evenly in the microwave.
Baseline	I need a way to scrub the kitchen floor without getting on my hands and knees.
5.0	I need a way to scrub the floors without getting on my hands and knees
Baseline	My knees hurt when I am scrubbing the floor.
5.0	My knees hurt when I am scrubbing the floor.
Baseline	You can spray cleaners in your eyes
5.0	You can spray cleaners in your mouth
Baseline	Trying to figure our how long the trip will take.
4.88	Trying to figure out how long the trip will take.
Baseline	I need a way to lessen my anxiety when it comes to flying.
4.42	I need a way to lessen my anxiety on long drives.
Baseline	I need a way to find driving directions easier
4.26	I need an easier way to get driving directions for when we travel.

TABLE 6. False Positive Examples: Predicted Similarity is Too High

Similarity Score	Equivalent Rating	Need Statements: Cooking, Cleaning, Travel (top to bottom)
Baseline		I need a way to know if my rice is done.
4.7	0.1	I need a easy way to know if my steak is done.
Baseline		I need a easier way to clean the outside of my windows.
4.5	0.0	I need an easier way to clean my bathtub
Baseline		I need a way to guarantee that my luggage will arrive.
3.6	0.0	I need an easy way to pack my luggage.

TABLE 7. False Negative Examples: Predicted Similarity is Too Low

Similarity Score	Equivalent Rating	Need Statements: Cooking, Cleaning, Travel (top to bottom)
Baseline		I need a way to lift a partially cut open can lid out of the can.
2.8	1.0	I need a better way to open cans.
Baseline		I need a good way to clean the top of a ceiling fan.
3.1	1.0	Ceiling fans are difficult to clean.
Baseline		I need a convenient way to make a checklist of things to pack
2.8	0.9	I need a better way to pack when traveling.

The specific ability to detect uniqueness (rather than duplication) may also have wide application in open innovation data management and supports future work in this area. In resource-constrained situations, an organization may have the ability to evaluate a set maximum number of options. STS rankings create the potential to determine a cutoff score based on the required final count of statements with the lowest similarity to each other. These statements would potentially allow rapid exploration of the entire data set. A further application of STS may then seek the similar variations of any high-quality statement from this exploration set. Lowest similarity scores may be exploited to find tacit or latent needs or novel ideas–those that are rarely articulated.

4.2 Reduced Resources for Automated Methods

The approximate computation time for rating the complete set of need pairs for each topic was 1 day. The computation times for larger sets would be limited only by processing speed. While previous examples, such as a 3 month Cisco project, likely performed assessments beyond only duplication, automated methods would likely compare favorably for this specific step.

4.3 Potential for Future Increases in Accuracy

The accuracy of true positives in predicted scores decreased considerably when a cutoff score reflects similar, but not equivalent sentences. While this increased subtlety may be challenging for current technology, human raters also demonstrated poor consensus in this region, and the continuing attention dedicated to NLP research may soon reduce the gap between prediction and human ratings. In this study alone, comparing 2012 to 2013, the results demonstrated a significant performance increase. Our best results for the 2012 algorithm required specialized tuning to an open-innovation application. The 2013 algorithm achieved almost equivalent performance with no manual tuning to our application area. The accuracy of STS may be further improved with application-specific development. For example, in some in-

stances the predicted scores indicated A=B, B=C, but A≠C, indicating this logic structure is not accounted for.

In addition, while STS algorithms are appropriate for statements relating to common consumer goods and services, similar NLP methods dedicated to specialized language, such as clinical vocabulary, are also enjoying research interest and the potential for ongoing improvements [19]

4.4 Evidence Against Fraud or Malicious Use

Table 5 includes some examples of exact duplication; however, exact duplicates are rare, and there is no evidence of widespread malicious submission (e.g. for increased pay) even when compared to sets of statements shown as the shared need stimulus (data not shown in results). One example in particular was traced to the same user submitting the same statement in rapid succession; however, after also submitting other unique needs. One potential explanation is inadvertently clicking the submit button multiple times.

4.5 Limitations and Future Work

Rates of duplication may be dependent on the specificity of the topic. While the three topics used in this study resulted in similar rates of unique entries over a wide range of group sizes, these topic were intentionally chosen to be broad enough to be relevant to a large pool of recruited participants. Initial expectations were to see first suggestions that are obvious and hence often similar; however, the quantity of submitted data and the degree of uniqueness exceeded expectations. A more narrow focus with similar group sizes will likely increase duplication; although this may be desirable as it indicates saturation in the data. Recruiting larger groups for broad topics is also possible; however, this may generate quantities of data that require larger computational resources if evaluated post data collection. However, future developments of real-time algorithm implementation would distribute computational analysis over a longer period. In this scenario, each new need would be immediately compared to all previously submitted needs (either all needs or filtered for only unique baseline entries) and processed at the time of submission. NLP algorithms are highly amenable to such use.

STS will only be applicable to submitted data in text form. This provides further justification for use with text-based need descriptions; however, idea submissions incorporating a visual component, such as a sketch, would not be suitable. Also, assessing a large pool of open innovation data specifically for similar and unique subsets is only one component of managing the data. A similarity rating does not imply whether the submission is of high quality (e.g. important to users, representing a market opportunity) and should be pursued further; however, the rating may facilitate later steps by reducing redundant evaluations and simplifying initial explorations of the space. Evaluating the quality of ideas has been thoroughly studied [20]. Research relating to the quality of need statements is more limited [21] and is further motivated by the results of the current work.

Only two algorithms were tested for this study. Candidates were chosen based on existing data suggesting superior performance over many other candidates; however, the specifics of comparing need or idea statements might result in outcomes where other algorithms produce better results. In addition, because the UMBC system source code was not publicly available, there was no way to retrain the system with new task-specific data. Therefore, UMBC does not use the same training data as the final TakeLab system, introducing an additional unknown in this comparison.

5 CONCLUSION

These results support the use of STS algorithms for automatically assessing needs-based open innovation textual data to facilitate comprehensive evaluations of quality. Automated analysis via STS algorithms can scale to the large volumes typical in open-innovation data and overcome a significant impediment to its widespread use: the bottleneck of prohibitive human resource costs to process such data. The accuracy of this method is promising, particularly given the low consensus levels of manual human ratings. The correlations of predicted similarity values to human ratings were generally high (up to .85) and indicated that specialized tuning of the algorithm to need statement applications is beneficial but not required. The accuracy of a similarity rating for statements with equivalent meaning approached .75, and the decrease in accuracy for ratings of lower similarity is partially attributed to a lack of consensus in human ratings for sentence equivalency. The improvement in results for non-tuned algorithms from 2012 to 2013 provides further support that NLP research will lead to ongoing improvements relevant to this application.

ACKNOWLEDGMENT

Thanks go to Ted Pedersen, PhD at the University of Minnesota Duluth for valuable early suggestions and direction relating to candidate NLP algorithms. Lushan Han at UMBC and Mladan Karan at the University of Zagreb for valuable assistance during algorithm testing.

REFERENCES

[1] Brabham, D. C., 2008. "Crowdsourcing as a Model for Problem Solving an Introduction and Cases". *Convergence: The International Journal of Research into New Media Technologies,* ***14***(1), pp. 75–90.

[2] Poetz, M. K., and Schreier, M., 2012. "The Value of Crowdsourcing: Can Users Really Compete with Profes-

sionals in Generating New Product Ideas?". *Journal of Product Innovation Management,* ***29***(2), pp. 245–256.

[3] Boudreau, K. J., and Lakhani, K. R., 2013. "Using the Crowd as an Innovation Partner". *Harvard Business Review,* ***91***(4), pp. 60–69.

[4] Enkel, E., Gassmann, O., and Chesbrough, H., 2009. "Open r&d and open innovation: Exploring the phenomenon". *R&D Management,* ***39***(4), pp. 311–316.

[5] Westerski, A., 2013. "Semantic Technologies in Idea Management Systems: A Model for Interoperability, Linking and Filtering". PhD thesis, Universidad Politécnica de Madrid (UPM), Madrid.

[6] Jouret, G., 2009. "Inside Cisco's Search for the Next Big Idea.". *Harvard Business Review,* ***87***(9), pp. 43 – 45.

[7] Cooper, R., and Edgett, S., 2008. "Ideation for Product Innovation: What are the Best Methods?". *PDMA Visions Magazine,* ***1***(1), pp. 12–17.

[8] McCandless, M., Hatcher, E., and Gospodnetic, O., 2010. *Lucene in Action: Covers Apache Lucene 3.0*. Manning Publications Co., Greenwich, CT, USA.

[9] Westerski, A., Iglesias, C., and Garcia, J., 2012. "Idea Relationship Analysis in Open Innovation Crowdsourcing Systems". In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*. Pittsburgh, PA, USA, Oct, pp. 289–296.

[10] Chowdhury, G. G., 2003. "Natural Language Processing". *Annual Review of Information Science and Technology,* ***37***(1), pp. 51–89.

[11] Cambria, E., and White, B., 2014. "Jumping NLP Curves: A Review of Natural Language Processing Research". *Computational Intelligence Magazine, IEEE,* ***9***(2), May, pp. 48–57.

[12] Agirre, E., Diab, M., Cer, D., and Gonzalez-Agirre, A., 2012. "Semeval-2012 task 6: A Pilot on Semantic Textual Similarity". In *Proceedings of 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*. Montréal, Canada, pp. 385–393.

[13] Schaffhausen, C., and Kowalewski, T., 2015. "Large-Scale Needfinding: Methods of Increasing User-Generated Needs from Large Populations". *Journal of Mechanical Design*. In Press, doi 10.1115/1.4030161.

[14] Faste, H., 2011. "Opening "Open" Innovation". In *Proceedings of the 2011 Conference on Designing Pleasurable Products and Interfaces*, DPPI '11. ACM, New York, NY, USA, pp. 54:1–54:8.

[15] Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., and Guo, W., 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*. Atlanta, Georgia, USA.

[16] Šarić, F., Glavaš, G., Karan, M., Šnajder, J., and Dalbelo Bašić, B., 2012. "Takelab: Systems for Measuring Semantic Text Similarity". In *First Joint Conference on Lexical and Computational Semantics (*SEM)*. Association for Computational Linguistics, Montréal, Canada, 7-8 June, pp. 441–448.

[17] Han, L., Kashyap, A., Finin, T., Mayfield, J., and Weese, J., 2013. "UMBC EBIQUITY-CORE: Semantic Textual Similarity Systems". In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task*. Atlanta, Georgia, USA, pp. 44–52.

[18] Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T., 2005. "ROCR: Visualizing Classifier Performance in R". *Bioinformatics,* ***21***(20), pp. 3940–3941.

[19] Pradhan, S., Elhadad, N., Chapman, W., Manandhar, S., and Savova, G., 2014. "SemEval-2014 Task 7: Analysis of Clinical Text". In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, Vol. 199. Dublin, Ireland, pp. 54–62.

[20] Dean, D. L., Hender, J. M., Rodgers, T. L., and Santanen, E. L., 2006. "Identifying Quality, Novel, and Creative Ideas: Constructs and Scales for Idea Evaluation". *Journal of the Association for Information Systems,* ***7***(10), pp. 646–698.

[21] Ulwick, A. W., 2002. "Turn Customer Input into Innovation.". *Harvard Business Review,* ***80***(1), pp. 91–7.