Natural Language Processing for scoring open-ended questions: a systematic review

Rutger Kerkhof

University of Twente
PO Box 217, 7500 AE Enschede
the Netherlands

r.g.kerkhof@student.utwente.nl

ABSTRACT

In education, the performance of students is measured using summative tests, which often consist of a combination of open- and close-ended questions. The latter can be automatically scored for almost a century, though, open-ended questions are still scored manually, which is both time- and resource-intensive work. This required effort can be severely reduced by automating this task, whilst also ensuring unbiased grading. This study will propose the state-of-the-art on automatic scoring of open-ended questions, by performing a systematic literature review, based on the guidelines proposed by Kitchenham. First, (pre-)processing techniques will be discussed, especially evaluating semantic similarities. Then, unsupervised machine learning techniques are considered to analyze the processed data. Finally, all found techniques will be compared, to determine how the state-of-the-art system for scoring open-ended questions should look.

Keywords

Natural Language Processing, scoring open-ended questions, semantic similarity, unsupervised learning, systematic review

1. INTRODUCTION

Summative tests are a useful and popular measure to assess students' knowledge on a topic. Traditionally, all types of questions had to be scored manually which is a very time-consuming process [15]. With the introduction of the first automatic multiple-choice test scoring machine, in 1937 by IBM, a new – and more efficient – era began [5]. Though, automatic scoring has never been as much of a success for open-ended questions. This is due to the difficulty of understanding natural language, as explained by Burrows et al. [1]. Burrows continues that past efforts have generally been ad-hoc and non-comparable. Current solutions require much development time or result in inconsistencies.

So, it is important to keep researching and improving this field, mainly to save human resources. Besides, automated scoring is also likely to be more objective [10]; graders can be biased, inter alia, by fatigue, the student's name, or the relationship to this student. A computer, on the other hand, will score comparable answers similarly, regardless of any such characteristics. Even though closed-ended questions can already be automatically scored, it is also important to not only assess students using this type of question. Since one can only measure so much of a student's comprehension by exclusively asking close-ended questions [13].

This study will focus on providing an overview of the ways that open-ended questions can be scored. The open-ended questions reviewed in this study are characterized by their length (one sentence to one paragraph) and focus. This focus is on whether the content and meaning of the given answer match the expected answer. This is completely different from essay questions, a type of open-ended question, which is longer than one paragraph and mostly focusses on the structure and grammar [1]. Closed-ended questions, on the other hand, can range from multiple-choice questions to fill-the-gap questions, neither of which requires a natural language response but rather a static one. Meaning that the range of possible answers is limited, compared to the free-text answers on open-ended questions.

Classically, there are three basic learning paradigms, namely supervised learning, unsupervised learning, and reinforcement learning, as explained by Kwok et al. [8]. However, only unsupervised learning will be considered in this study, the other two will not be included. Using reinforcement learning for scoring open-ended questions has not been researched enough; a search on Scopus did not yield any results. Supervised learning does not augment the basic principle of saving human resources, which defeats the main goal of the automated scoring of open-ended questions. Supervised learning requires labeled data (human-scored questions in this case) to train the algorithm on. This means that when a question is altered or added, manual labeling needs to be done before they can be graded automatically. For rapid developing domains, like technology, this might bring a lot of additional work, since new developments can happen from year to year and lead to new (possible) questions. Unsupervised learning, on the other hand, does not require any labeled data and, thus, no additional manual work when new questions are added or altered. This also makes it easier and more approachable to start using or testing automated scoring. Therefore, unsupervised learning is the focus of this study.

There has been a lot of research towards the automatic grading of open-ended questions. However, no review is available specifically aimed at solving this problem using unsupervised machine learning. So, the aim of this review is to contribute to the developments in this field.

In section 2 the used methodology will be explained, along with the search process. In section 3 the selected literature will be discussed, to answer the sub-questions. Finally, the study will be concluded in section 4, by answering the main research question, and by mentioning the limitations of this study and future work.

33th Twente Student Conference on IT, July 3rd, 2020, Enschede, The Netherlands.

2. METHODOLOGY

This study will follow the guidelines for performing a Systematic Literature Review (SLR), as proposed by Kitchenham [6]. These guidelines are for developing a Review Protocol and consists of five steps, of which the first three will be followed in section 2.2, 2.3, and 2.4, respectively. The final two proposed steps (data extraction and data synthesis) will not be explicitly described and performed, considering the small scale and scope of this study; this is also suggested by Kitchenham. Data extraction and synthesis are the process of extracting and summarizing information from the selected studies.

2.1 Research Questions and Scope

The goal of this study is to find which techniques are most suitable for the automatic scoring of open-ended questions. This will be achieved by answering the following research question:

RQ1 What is the state-of-the-art on automated scoring of open-ended questions?

To answer this question, different techniques will be addressed for both the pre-processing of the data, as well as analyzing it. To do this systematically, the following sub-questions will first be answered.

The data to be processed consists of answers to open-ended questions, given by students on summative tests. This natural language will first be pre-processed using some traditional techniques, like stop word removal and stemming. Besides those, some more interesting techniques will be discussed, to check semantic similarity. This gives the first sub-question.

RQ1.1 Which (pre-)processing techniques can be used for assessing open-ended questions?

Now that it is clear how the data will look, different techniques for analyzing it will be considered. As explained in the introduction, only unsupervised machine learning will be discussed. Supervised machine learning still requires a lot of manual work (labeling training data) and automated scoring using semi-supervised leaning has not been researched enough. This gives the second sub-question.

RQ1.2 Which unsupervised learning techniques can be used for analyzing open-ended questions?

To find out which of the proposed techniques in RQ1.1 and RQ1.2 are most promising, a comparison will be done in the third sub-question, by checking similarities, pros, and cons.

RQ1.3 How do these techniques compare to each other?

2.2 Identification of Research

The first step in an SLR is to determine and follow a search strategy. The main digital library that will be used is Scopus since this database ensures high-quality papers and claims to deliver the broadest overview of data and literature, across all research fields [4]. However, when specific work found using snowballing (explained in section 2.5) is not available on Scopus, then Google Scholar will be used. Mendeley will be used to manage and import all references.

2.3 Study Selection

The relevance of found studies needs to be assessed. This can be done through inclusion and exclusion criteria, which are based on the research question. First, an initial search will be performed on Scopus, using three queries related to each other with an AND operator. These queries will search for the specified terms in the **title, abstract, and keywords**:

1. "Natural Language Processing" OR nlp OR "natural language"
2. (open AND question) OR "short answer"
3. grading OR scoring OR marking OR asses* OR exam

With these three base queries, Scopus returns **151** documents. In figure 1, this initial search, and the following two steps of the inclusion and exclusion process are depicted, including how many documents are left after each respective step. The next step is to further filter these documents, using the following three criteria:

Crit1 Whether it concerns processing of English text

Crit2 Whether it concerns automatic grading of open-ended questions

Crit3 Whether it concerns unsupervised machine learning

Whether these criteria are satisfied or not, will be assessed by reading the **title and abstract** of all papers. If one of the criteria is not satisfied, the paper will not be included in the review. In case it is not clear whether the paper satisfies a criterion, it will be accepted to be inspected more thoroughly in the next step (quality assessment). After all papers have been assessed using these three criteria, **28** documents are left.

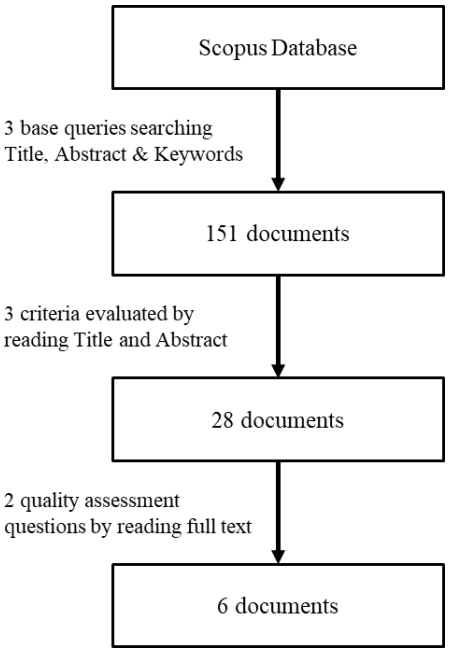

Figure 1. Documents left after performing the initial search, evaluating criteria, and quality assessment, resp.

2.4 Study Quality Assessment

In the quality assessment, more specific/in-depth inclusion and exclusion criteria are considered, to assess the quality of the chosen studies. At the same time, the ambiguous cases which were accepted in the previous step will be checked more thoroughly.

Kitchenham explains that the goal of this step is to apply extra inclusion and exclusion criteria, to minimize bias and maximize internal and external validity. However, it is also suggested to, instead, generate a list of quality assessment questions keeping the study's context and research questions

into account. The latter will be done here; the quality assessment questions are:

QA1 Are all criteria stated in section 2.3 satisfied?

QA2 Are the Text Mining (pre-processing) and/or Machine Learning techniques explicitly mentioned and sufficiently explained?

To answer these questions, the remaining 28 documents will be scanned and, if necessary, the **full text** will be read to ensure the document is both relevant and of high quality. Like the criteria above, studies will only be used if they satisfy both quality assessment questions. After all papers have been assessed using these questions, **6** documents are left to be used in this review.

Table 1. Studies selected for this review

#	Title	Ref.
1	The eras of automatic short answer grading	[1]
2	Automarking: Automatic assessment of open questions	[2]
3	Auto-assessor: Computerized assessment system for marking student's short-answers automatically	[3]
4	Vector based techniques for short answer grading	[9]
5	Automatic short answer grading and feedback using text mining methods	[16]
6	Automatic Coding of Short Text Responses via Clustering in Educational Assessment	[18]

2.5 Snowballing

After performing the SLR, there might still be a need for extra papers to get a deeper insight into some techniques. These papers will be found through a process called backward snowballing, as explained in [17]. This is the process of identifying new papers to include, based on the reference list of already selected studies. Before a study is used in this review, it will first be evaluated using the above-mentioned criteria and quality assessment questions. The study is only used if all criteria are satisfied. In table 2 the papers included through snowballing can be found.

Table 2. Papers included through backward snowballing

#	Authors	Referenced in
1	Manning et al. [11]	Magooda et al. [9]
2	Mohler and Mihalcea [12]	Burrows et al. [1]
3	Kolb [7]	Magooda et al. [9]
4	Pedersen et al. [14]	Burrows et al. [1]

3. RESULTS

In this section, the findings of the systematic literature review, and answers to the sub-questions will be presented. Below, each sub-question is discussed in its sub-section.

3.1 (Pre-)processing using Text mining

Which (pre-)processing techniques can be used for assessing open-ended questions?

To analyze the student answers, Natural Language Processing (NLP) techniques are required. These can generally be split into five broad categories: lexical {1}, morphological {2}, semantic {3}, syntactic {4}, and surface {5}. The former and latter two are quite basic and trivial techniques, to the point that in many studies their use is not explicitly mentioned, just implied [1]. Below some examples of these basic pre-processing techniques will be briefly explained [2], [11, ch.2]. The number {x} after each technique indicates in which of the five categories it belongs.

- Stop words can be removed {1} as they do not value or have no meaning, examples of such words are 'the', 'a', 'an', 'of' and 'to'. In the case of scoring open-ended questions, words used in the question sentence might also be considered to not add value, since they are common to repeat [16].
- Stemming {2} brings a word back to its word stem; the affix of the word is cut off and verbs are put in infinitive form.
- Lemmatization {2} brings a word back to its lemma, keeping the context into account. This will probably give a better result for ambiguous words like 'saw', by looking at whether it is a verb or a noun.
- Tokenization {4} is the process of breaking a sentence up into tokens (words), while at the same time removing symbols {5} that do not add value or have no meaning, such as interpunction, capitalization, hyphens, and apostrophes. A common addition is to give all tokens a part-of-speech (POS) tag {4}, such as noun, verb, adjective, and adverb [3].

The use of semantic is, on the other hand, less obvious and trivial. These techniques focus on semantic relations and similarities between words or sentences [1], [3], [9], [18]. The similarity is expressed using a real number between 0 ~ 1, where 0 means no similarity and 1 an exact match in meaning. These measures of similarity can be used as features for classifying or clustering answers [9], this will be discussed in section 3.2.

The two most popular methods to check the (semantic) similarity of words are corpus-based and knowledge-based [9]. Corpus-based methods use the statistical information collected by processing large corpora and does, thus, not require a pre-built knowledge source. Knowledge-based similarity uses an external lexical database containing sets of cognitive synonyms, named 'synsets' [3]. Below, some examples of both similarity methods will be explained, and at the end string similarity will also be explained.

3.1.1 Corpus-based similarity

In the selected literature, three different types of corpus-based similarity measures were used and explained. All three methods used 'word vector representation' (known as word embedding) and measured the similarity between words by calculating the distance between these vectors. These word vectors exist in a high dimensional space, where each dimension holds semantic or syntactic features of words [9]. For example, the word vectors of 'fantasy' and 'imagination' have a small angle since their (semantic) meaning is similar [18]. To calculate the distance between vectors, **distance measures** are used. Some popular measures are cosine distance, Euclidian distance, and Manhattan distance [9].

Latent Semantic Analysis (LSA) is a popular and influential approach that uses a 'bag of words' (BOW) to evaluate the similarity of words [2], [3]. In BOW, the order of words is ignored, only the number of occurrences is taken into account [11]. So, when BOW is used, 'John is quicker than Mary' is identical to 'Mary is quicker than John'. LSA needs to use a domain-specific corpus, to reach its full potential. This can, for example, be done by starting at a strongly related article and then add incoming and outgoing links from this article to the corpus [18].

Explicit Semantic Analysis (ESA) is based on a manually structured text corpus, meaning documents, like in Wikipedia articles, are included as intact entities [18]. This means that the dimensions of the vectors are directly equivalent to abstract concepts [12]. So, in case of Wikipedia articles, each article represents a concept in the ESA vector. For example, the word 'fantasy' is represented by a vector comprised of over one million weights (as many weights as articles), each indicating the relatedness between the article and term.

Extracting distributionally similar words using co-occurrence (DISCO) is based on the assumption that words with a similar meaning occur in similar contexts [7]. Distributional similarity is a relation between words, while semantic similarity is a relation between concepts. DISCO scans the corpus using a variable window for counting co-occurrences to keep the context into account [9].

3.1.2 Knowledge-based similarity

WordNet is the most popular source of synets, containing over 150,000 lexical and conceptual meanings, functioning a lot like a thesaurus [3]. Words are categorized based on their POS tag (nouns, verbs, adjectives, and adverbs). Below, some of the word-to-word similarity and relatedness metrics, that can be found in the WordNet::Similarity package [7], [12], [14], will be discussed. **Semantic similarity** is a narrow concept matching words with a similar meaning, like 'palm' and 'tree'. **Semantic relatedness** is a broader concept matching words by lexical relations, like 'palm', 'leaf', and 'coconut' [7].

Mohler and Mihalcea [12] compared eight different knowledge-based measures. Six of these, which were also used and/or explained in [9] or [14], will be shortly explained below. The other two (shortest path and Resnik) were not explained or used in another selected study and, thus, not included. Of these six, four are similarity measures, and the other two measure the relatedness of two words.

Leacock & Chodorow similarity tries to find the shortest path between two concepts by counting nodes. The shorter this path, the higher the semantic similarity is of the two concepts [12], [14].

Lesk checks the relatedness of two words by counting the shared terms found in the WordNet definition. It is based on a solution for word sense disambiguation [9], [12], [14].

Wu & Palmer measures the similarity of two concepts by using the depth of both given concepts in the WordNet taxonomy, and the depth of the least common subsumer (LCS). The depth means the distance to the root node, and the LCS of two concepts is the most specific concept that is an ancestor of both [12], [14].

Lin similarity is built on Resnik's similarity measure, which returns a value, the information content (IC), based on the occurrence of the LCS of two concepts in a large corpus. Lin added a normalization factor onto this measure by including the IC of both input concepts [12], [14].

Jiang & Conrath similarity uses the same principle as Lin's – IC of both input concept and their LCS – to find the similarity of two concepts but calculates the output value using a different formula [9], [12], [14].

Hirst & St. Onge checks the relatedness of two terms by checking the relations in WordNet. Examples of these relations are 'is-a', 'type-of', and 'part-of'; these relations are directional. The relatedness of two terms is based on the length of the path and the number of times that the direction has to change [12], [14].

3.1.3 String similarity

Besides these two forms of semantic similarity, two strings can also be compared regardless of their meaning. The similarity score will be based on the minimum number of operations required to convert one string to the other [9]. The most popular algorithm for doing this is 'Levenshtein Distance', which transforms the string through addition, removal, and substitution. String similarity can be useful in case a word has been written incorrectly since misspelled words cannot be used in semantic similarity. Another case is when stemming or lemmatization is used, to relate words like 'poni' (the stem of 'ponies') to 'pony' [11, ch.2].

3.2 Unsupervised learning

Which unsupervised learning techniques can be used for analyzing open-ended questions?

Unsupervised learning is one of three main Machine Learning paradigms, of which the main characteristic is that all data used is unlabeled [11, ch.16]. This study only focuses on this paradigm since the goal is to reduce as much manual labor as possible. Hence, labeling data (in this case: scoring answers), should be prevented if possible, as this is generally done manually. Another benefit of unsupervised learning is that it can be applied to any type of domain; whether it changes rapidly, like technology, or is static and set in stone, like law [18].

The focus will be on the most common form of unsupervised learning, namely clustering. Clusters are groups created based on more similarity between answers [11, ch.16]. So, answers in the same cluster are more similar to each other than those in other groups. This similarity is based on the answer's features, like all measures mentioned in the previous section.

The number of clusters can be based on either theoretical or empirical knowledge [16], [18]. The most popular method to decide on this is the elbow method, other methods are average silhouette and gap statistic method. Based on empirical knowledge means deciding on the number of manually, case-specific, or based on personal preference. Answers can be clustered in several ways, for example, using an interval (divide a 1~10 scale into pieces) or an ordinal (excellent/mixed/weak or pass/fail) scale.

Based on the selected literature it can be concluded that there are two main clustering methods used for scoring open-ended questions. Below, we will first discuss k-means clustering and then hierarchical clustering.

***k*-means clustering** [11, ch.16], [16] is one of the most popular methods for clustering data. In this method, the goal is to minimize the average squared Euclidean distance of *documents* (student's answers) to the *centroid* (center of the cluster), referred to as RSS. At the start of the *k*-means algorithm, a pre-set number of clusters will be created: *k*. Each cluster is represented by a centroid, which will be randomly placed in space. The algorithm then moves the centroid in order to minimize the average distances. Then, documents are reassigned to the cluster of the closest centroid, followed by each centroid recalculating its location based on the new cluster of documents. These two steps are iteratively repeated until a stopping criterion has been reached: the maximum number of iterations, clusters did not change, centroids did not change location (significantly), the average distance falls below the threshold. This kind of method is also called *flat clustering*, as there is no explicit structure connecting all clusters, as opposed to *hierarchical clustering*.

Hierarchical clustering [11, ch.17]**,** [18] builds a hierarchy of clusters, as opposed to *flat clustering* methods. One method is to cluster agglomerative, meaning that first all answers are considered to be their own cluster. Then, the distance between all clusters is calculated and the two that are closest together, are merged. Then the distances are recalculated, to merge the closest ones again, etc. This process is repeated until the desired number of clusters has been reached. The reversed process is called divisive, which approaches the problem top-down. Even though hierarchical clustering does not require a predefined number of clusters, one can still decide to cut it at a predefined point. This can be done using several criteria, like a predefined level of similarity, the minimal average distance (penalizing additional clusters), or at a specified number of clusters.

3.3 Comparing all techniques

How do these techniques compare to each other?

All techniques found in the previous section will be compared here by checking similarities and evaluating the pros and cons of each technique.

3.3.1 Pre-processing the natural language

Since most pre-processing techniques are considered trivial by researchers, there are no explicit comparisons between these techniques. One category of pre-processing techniques that however is being discussed is morphological. Hence, this will be discussed and compared below.

Both stemming and lemmatization try to bring words back to their common base form, for normalization purposes, as their semantic meaning is (often) similar [11, ch.2]. Stemming achieves this by 'blindly' chopping off the end of the word, hoping to be correct. Lemmatization, on the other hand, uses a vocabulary, morphological analysis, and considers the context (POS-tags), to bring a word back to its lemma.

However, using either stemming or lemmatizations is not unanimously considered useful or positive. Some argue it is important and is valuable to add [3], [9], [18], others doubt how useful it is and whether it outweighs the downsides (efficiency and incorrect changes) [11, ch.2]. However, another downside of applying these morphological techniques is limiting the tool to be used to supported languages only [7]; which is not in the scope of this study but is important to keep in mind.

3.3.2 Corpus-based semantic similarity

Based on the literature found for corpus-based techniques, LSA outperformed ESA [12], [18]. Despite not knowing the true reason for the difference in performance, some plausible reasons were given:

- LSA is optimized for a single domain (expert corpus), whereas ESA used a more general corpus (carrying irrelevant information). It was also verified by [12] that the type of corpus impacted the results, this was tested by comparing the performance of the same LSA algorithm; one using a generic corpus and the other using a domain-specific one. However, when ESA uses a domain-specific corpus, it shows a decrease in performance, compared to the one using the generic corpus.
- It is expected that the bag-of-words had a smaller effect on ESA than on LSA. A possible cause for this might be that the word vector did not drastically alter based on it, therefore making the overall effect smaller.

Despite the better performance compared to ESA, LSA also has some downsides that should be considered [18]. A reasonably big corpus is required to achieve high efficiency and, depending on the domain, this corpus might need to be regularly updated. Secondly, LSA does not consider the order in which words appear [3]; this causes LSA to consider the following two sentences equivalent: "the boy stepped on a spider" and "the spider stepped on a boy".

In addition to the more popular LSA and ESA, there is also DISCO. According to evaluations done by the creator of the algorithm, Kolb [7], DISCO has a higher correlation with semantic similarities derived from WordNet than LSA. Besides, Kolb also showed that DISCO had a higher correlation with semantic relatedness judgments made by humans, compared to (knowledge-based) WordNet methods. It did however get outperformed by LSA on this. The DISCO algorithm got used in [9], where it got preferred over both ESA and LSA; no explanation for this was given.

Three different distance measures were mentioned: the cosine, Euclidean, and Manhattan distance [9]. All three distance measures performed equally well [18], however, the cosine distance might be favorable. This is the case since it does not consider the length of a vector, which is a representation of the term frequency.

3.3.3 Knowledge-based semantic similarity

The main problem that occurs when using an external lexical database, like WordNet, is that it does not always cover all desired aspects, like language and domain-specific terms [7], [12]. For corpus-based similarity, this is not as much of a problem as you can use a corpus in the desired language or about a specific topic. The language is, however, not as big of an issue for this study since only English answers are considered. The limited domain-specific coverage is, on the other hand, quite a big constraint since most questions will require a deep insight into the domain.

Six different knowledge-based measures for semantic similarity were discussed, of which two focused on measuring the semantic relatedness of concepts. All six measures have the same goal: finding out how two words are connected; either through similarity ('palm' and 'tree'), or relatedness ('palm' and 'leaf').

There is, however, no unequivocal comparison on their correlation to human grading. Both [7] and [12] attempted to measure this, but their results regarding the performance are divergent, and for some measures even contradicting. The two are compared in table 3; the biggest variance being in the measure by Jiang & Conrath (jcn) and the smallest variance in the measure by Wu & Palmer (wup). The other measures are represented as follows: Leacock & Chodorow (lch), Lesk (lesk), Lin (lin), Hirst & St. Onge (hso).

Table 3. Correlation of knowledge-based measures measured by Kolb [7] **and Mohler** [12]

	lch	**lesk**	**wup**	**lin**	**jcn**	**hso**
[7]	0.35	0.21	0.30	0.30	0.23	0.35
[12]	0.2231	0.3630	0.3366	0.3916	0.4499	0.1961

3.3.4 Unsupervised learning

k-means clustering and hierarchical clustering both have the same goal, but both achieve it differently. The main difference between these two algorithms is that the former is categorized as a flat clustering algorithm and the latter as a hierarchical clustering algorithm. Some differences that this induces will be discussed below.

The main reason to choose flat clustering over hierarchical clustering is that the former is more efficient [11, ch.16-17]. This would mean that the algorithm takes less time to finish clustering all answers and return it as output.

The lack of structure is a problem as it is not easy to see how or when an answer got added to a specific cluster, which could be beneficial if the teacher would want to review or adjust the algorithm and its choices [11, ch.17]. This human interference is possible for hierarchical clustering but not for flat clustering. If you decide on a different number of clusters, flat clustering requires you to rerun the entire algorithm. Hierarchical clustering, on the other hand, allows you to go one step back or further, depending on whether you want extra or fewer clusters [18]. This could be beneficial in case a teacher feels the need to finetune the outcome of the algorithm.

The fact that flat clustering is non-deterministic means that every time you run the algorithm (on the same set of answers), you might get a slightly different answer [11, ch.17], [16]. This is caused by the centroids being initialized at a random location in space.

In addition to that, many researchers believe that clusters made hierarchically are better than those produced through flat clustering; there is, however, no proof for this [11, ch.17].

So, to decide whether to use flat or hierarchical clustering, a choice needs to be made between efficiency (flat) and all the above-mentioned advantages (hierarchical).

4. CONCLUSION

In this study, a systematic literature review was performed to find the current state-of-the-art on the automated scoring of open-ended questions. This study consisted of three main parts: pre-processing data, processing data (adding features), and clustering data. The focus of this study is on unsupervised learning, as this paradigm of machine learning requires the least amount of human input, which is the main goal. Clustering is considered to group semantically similar answers, decided by using both corpus- and knowledge bases semantic similarity measures. Often, when comparing two techniques the main question is whether efficiency could be sacrificed for higher performance. However, in the case of automating the scoring process, efficiency is not the main concern. A few extra seconds or minutes of computer processing is neglectable compared to manual grading. In addition to that, a higher performance might make the computer's judgments closer to those of humans. Therefore, performance will always be considered more important than efficiency, in this study.

Based on all reviewed literature, it can be assumed that most **pre-processing techniques** can be used, as no explicit objections were mentioned. The use of some morphological techniques – i.e. stemming and lemmatization – is debatable. However, the consensus is that they (marginally) outweigh the disadvantages and that lemmatization is less efficient but more effective than stemming. So, based on the literature, the best result of pre-processing is achieved when all techniques are used: lexical, morphological, syntactic, and surface.

More interestingly, techniques measuring the semantic similarity between two terms were also discussed, **processing techniques**. These were split into two distinct categories, corpus-based and knowledge-based. The former requires a big corpus from which it establishes semantic relations itself, while the latter uses a pre-built knowledge source that already stored all these relations, like WordNet, and can be used as an external lexical database.

Both corpus- and knowledge-based techniques result in a value between $0 \sim 1$, depending on how similar the two terms are. This measure of similarity can be used as a feature for the clustering and, therefore, all discussed measures should be included, weighted according to their performance, to attain a more reliable aggregate measure on similarity. The corpus-based techniques that were reviewed (ESA, LSA, and DISCO) used the distance between the vector representation of the words to calculate similarity, while the knowledge-based techniques used formulas with pre-determined characteristics. For this, using the cosine distance is considered favorable as this does not include the length (term frequency) of the vector, which is regarded as irrelevant, for measuring semantic similarity. The reviewed literature deemed ESA the worst of the three, meaning ESA should be given a lower weight. No clear and unbiased comparison between LSA and DISCO was done, so further research should be done to find out whether there is a notable difference in their performance, to allocate weights accordingly. For the knowledge-based techniques, there is no clear consensus on which is the best or worst. Further research is required to establish an estimate on the performance, to give a higher or lower weight to this technique.

Finally, **unsupervised clustering** methods were discussed and compared. Two popular methods of clustering were discussed: *k*-means and hierarchical clustering. The main difference in performance is that the former is more efficient, but the latter is regarded as more accurate. Besides being more accurate, hierarchical clustering is also easier to adjust or review manually because all steps are stored in the hierarchy.

So, the state-of-the-art system found through this systematic review consists of the following techniques and methods:

First, tokenization is applied to the natural language, giving every token a POS tag. Then, this data is further prepared by removing stop words and symbols (interpunction, capitalization, hyphens, etc.), to then perform lemmatization. Now, all noise is removed, and the leftover text has been normalized. The same steps are applied to the teacher's answer, to do a similarity check can be done between the student's and the teacher's answer. This is done using all the before-mentioned corpus- and knowledge-based measures and will all be weighted according to their performance or accuracy. Finally, hierarchical clustering will be used to cluster answers based on these features.

4.1.1 Discussion

For the first sub-question (RQ1.1), it was found that most techniques for pre-processing data are not explicitly mentioned, explained, and discussed. A plausible reason for this is that researchers find it trivial to use these techniques and, thus, do not find it interesting to discuss them. However, for techniques that did not have unanimous support, there were discussions, like stemming versus lemmatization and different similarity measures. This was, therefore, also the main aim of this question. For similarity measures, the consensus was to use several, as this might benefit the performance. This was, thus, also accepted in this review.

For the second sub-question (RQ1.2), there was quite a little amount of research found performing unsupervised learning techniques to solve this problem. The studies that did use this only used considered clustering and did not compare it to other forms of unsupervised learning. Hence, the focus of this question was on the use of unsupervised clustering.

For the third sub-question (RQ1.3), the previously found papers were used and the comparisons discussed in these papers were used to answer this question. These comparisons and considerations were used to come to a conclusion, to answer the main research question (RQ1).

4.1.2 Limitations

Since this SLR was performed in a limited time frame, the reviewed literature was also limited. This, in combination with little to no prior knowledge about how to perform an SLR, caused a limited review where some interesting techniques could not be discussed. Some examples of these techniques are sentence embedding, unsupervised neural networks, and spectral clustering.

Secondly, since this is a review, no actual comparisons were done. This means that the proposed state-of-the-art system is a theoretical one. Testing and improving this (theoretical) system is considered future work. However, the proposed system does offer a solid start.

4.1.3 Future work

As mentioned above, this is only a theoretical state-of-the-art proposal and, thus, offers quite some room for future work.

The main goal of the future work should be to include methods and techniques that were left out of this study. These could first be compared to the proposed system in this review by using literature. However, it is also important to create this system and test it on real data. Below is a short list which should be included in future research:

- Exploring different techniques, inter alia, unsupervised neural networks and spectral clustering.
- Sentence embedding, this could either be combined with or replace word embedding. Many studies mentioned that most measures (in the WordNet::Similarity package) are already able to do sentence embedding, but no study compared this to the word embedding version.
- Weights for the different similarity measures should be improved based on their (positive) impact on the accuracy. This should be done after sentence embedding has been explored.

5. ACKNOWLEDGMENTS

I am very grateful to my supervisor Adina Aldea for her valuable feedback and help during this study.

6. REFERENCES

[1] S. Burrows, I. Gurevych, and B. Stein, “The eras and trends of automatic short answer grading,” *Int. J. Artif. Intell. Educ.*, vol. 25, no. 1, pp. 60–117, 2015, doi: 10.1007/s40593-014-0026-8.

[2] L. Cutrone and M. Chang, “Automarking: Automatic assessment of open questions,” in *Advanced Learning Technologies, ICALT 2010*, 2010, pp. 143–147, doi: 10.1109/ICALT.2010.47.

[3] L. Cutrone, M. Chang, and Kinshuk, “Auto-assessor: Computerized assessment system for marking student’s short-answers automatically,” in *Technology for Education, T4E 2011*, 2011, pp. 81–88, doi: 10.1109/T4E.2011.21.

[4] Elsevier, “How Scopus Works,” Mar. 10, 2015. https://www.elsevier.com/solutions/scopus/how-scopus-works/content (accessed Jun. 26, 2020).

[5] IBM, “Automated Test Scoring,” Feb. 11, 2011. https://www.ibm.com/ibm/history/ibm100/us/en/icons/testscore/ (accessed May 31, 2020).

[6] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering,” Durham, Jul. 2007.

[7] P. Kolb, “DISCO: A Multilingual Database of Distributionally Similar Words,” in *KONVENS*, Jan. 2008, pp. 5–12.

[8] J. T. Kwok, Z.-H. Zhou, and L. Xu, “Machine Learning,” in *Springer Handbook of Computational Intelligence*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 495–522.

[9] A. Magooda, M. A. Zahran, M. Rashwan, H. Raafat, and M. B. Fayek, “Vector based techniques for short answer grading,” in *FLAIRS 2016*, 2016, pp. 238–243.

[10] J. M. Malouff and E. B. Thorsteinsson, “Bias in grading: A meta-analysis of experimental research findings,” *Aust. J. Educ.*, vol. 60, no. 3, pp. 245–256, Nov. 2016, doi: 10.1177/0004944116664618.

[11] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press, 2009.

[12] M. Mohler and R. Mihalcea, “Text-to-text Semantic Similarity for Automatic Short Answer Grading,” in *12th Conference of the European Chapter of the ACL*, Mar. 2009, pp. 567–575.

[13] Y. Ozuru, S. Briner, C. A. Kurby, and D. S. McNamara, “Comparing comprehension measured by multiple-choice and open-ended questions.,” *Can. J. Exp. Psychol.*, pp. 215–227, Sep. 2013, doi: 10.1037/a0032918.

[14] T. Pedersen, S. Patwardhan, and J. Michelizzi, “WordNet::Similarity - Measuring the Relatedness of Concepts,” in *Demonstration Papers at HLT-NAACL*, Apr. 2004, pp. 38–41.

[15] J. Z. Sukkarieh and S. G. Pulman, “Information Extraction and Machine Learning: Auto-Marking Short Free Text Responses to Science Questions,” in *Supporting Learning through Intelligent and Socially Informed Technology*, Jul. 2005, pp. 629–637.

[16] N. Süzen, A. N. Gorban, J. Levesley, and E. M. Mirkes, “Automatic short answer grading and feedback using text mining methods,” in *Procedia Computer Science*, 2020, vol. 169, pp. 726–743, doi: 10.1016/j.procs.2020.02.171.

[17] C. Wohlin, “Guidelines for snowballing in systematic literature studies and a replication in software engineering,” in *ACM International Conference Proceeding Series*, May 2014, pp. 1–10, doi: 10.1145/2601248.2601268.

[18] F. Zehner, C. Sälzer, and F. Goldhammer, “Automatic Coding of Short Text Responses via Clustering in Educational Assessment,” *Educ. Psychol. Meas.*, vol. 76, no. 2, pp. 280–303, 2016, doi: 10.1177/0013164415590022.