Towards efficient scoring of student-generated long-form analogies in STEM

Thilini Wijesiriwardene[1,*], Ruwan Wickramarachchi[1], Valerie L. Shalin[1,2] and Amit P. Sheth[1]

[1] *AI Institute, University of South Carolina, Columbia, SC, USA*

[2] *Department of Psychology, Wright State University, Dayton, OH, USA*

Abstract

Switching from an analogy pedagogy based on comprehension to analogy pedagogy based on production raises an impractical manual analogy scoring problem. Conventional symbol-matching approaches to computational analogy evaluation focus on positive cases, and challenge computational feasibility. This work presents the Discriminative Analogy Features (DAF) pipeline to identify the discriminative features of strong and weak *long-form* text analogies. We introduce four feature categories (semantic, syntactic, sentiment, and statistical) used with supervised vector-based learning methods to discriminate between strong and weak analogies. Using a modestly sized vector of engineered features with SVM attains a 0.67 macro F1 score. While a semantic feature is the most discriminative, out of the top 15 discriminative features, most are syntactic. Combining this engineered features with an ELMo-generated embedding still improves classification relative to an embedding alone. While an unsupervised K-Means clustering-based approach falls short, similar hints of improvement appear when inputs include the engineered features used in supervised learning.

Keywords

Descriptive analogies, Analogical features, Analogy scoring, Long-form analogies,

1. Introduction

Analogical reasoning relies on the ability to draw on the relational similarities between two systems of objects in different contexts [1, 2, 3]. Analogies appear in several disciplines such as engineering design, scientific reasoning, and often in STEM education. However, the dominant pedagogical paradigm requires students to comprehend curated analogies. In this work, we are focusing on the evaluation of *student-generated* analogies in their first undergraduate biochemistry course.

Problem sets, specifically created to explore the underlying mechanisms of analogical reasoning, consist of visual and verbal analogies [4, 5]. Verbal analogies have two primary forms; analogical proportions and long-form analogies. Analogical proportions follow a four-term

ICCBR Analogies'22: Workshop on Analogies: from Theory to Applications at ICCBR-2022, September, 2022, Nancy, France

[*] Corresponding author.

thilini@sc.edu (T. Wijesiriwardene); ruwan@email.sc.edu (R. Wickramarachchi); valerie.shalin@wright.edu (V. L. Shalin); amit@sc.edu (A. P. Sheth)

0000-0001-8431-8443 (T. Wijesiriwardene); 0000-0001-5810-1849 (R. Wickramarachchi); 0000-0001-8135-2793 (V. L. Shalin); 0000-0002-0021-5293 (A. P. Sheth)

CEUR Workshop Proceedings (CEUR-WS.org)

format such as "*A* to *B* as to *C* to *D*" or *A*:*B*::*C*:*D* [6]. Recent work on computational analogy making focuses on analogical proportions [7, 8, 9]. Our interest here lies in *long-form* analogies consisting of a narrative/ description of a target unfamiliar situation/ system (for the context or lesson to be learned) using several sentences and a familiar source (base) [10, 3]. While the objects across the two descriptions differ, they employ similar *relations* between these objects. An example of a well-known long-form analogy is between the solar system (source) and the Rutherford-Bohr model of the atom (target) [11] where small objects revolving around a large central object provide relational similarity with the target. The solar system and the atom can each be described using several sentences. Parallels between these two systems can then be drawn, making the two descriptions analogous.

The atom-solar system analogy exemplifies the curated analogies in STEM textbooks. [12] has developed algorithms for evaluating correct or slightly incorrect long-form analogies. In [13] we solicited analogies from STEM students, with the expectation, relative to a comprehension exercise, that analogy production is both more engaging and allows students to employ existing familiar knowledge to scaffold the acquisition of new knowledge. No matter how pedagogically successful, manual scoring is impractical for an analogy production pedagogy. Production pedagogy elevates an analogy scoring problem for computational solution.

We aim to identify discriminative features between strong and weak, long-form *student generated* verbal analogies collected in a college biochemistry class (see Section 2.1) to support efficient computational scoring. To this end, we develop the Discriminative Analogy Features (DAF) pipeline.

We use a long-form analogy dataset, instructor-graded as strong or weak. We explore both supervised and unsupervised learning classifiers using vectors based on embeddings, engineered features and both. Given manually annotated data for a supervised learning classifier (i.e. SVM), we identify the discriminative features of strong and weak analogies.

We introduce DAF, a pipeline to identify the discriminative features of strong and weak analogies. We also introduce four feature categories – semantic, syntactic, sentiment, and statistical used in supervised learning to discriminate between strong and weak analogies. We show that "unique attribute count", a *semantic feature*, is the most discriminative when identifying between strong and weak analogies. Out of the top 15 discriminative features, most are *syntactic*. Unsupervised learning is unable to obtain comparable success, though it slightly improves with features corresponding to the above categories.

The rest of this paper is organized as follows: Section 2 introduces and describes the DAF pipeline and identifies the discriminative features. Section 3 presents the discussion with findings, insights, limitations, and future work subsections. Section 4 concludes the paper.

2. Discriminative Analogy Features (DAF) Pipeline

To identify discriminative features, we introduce the pipeline illustrated in Figure 1. In the subsequent subsections, we describe each pipeline component.

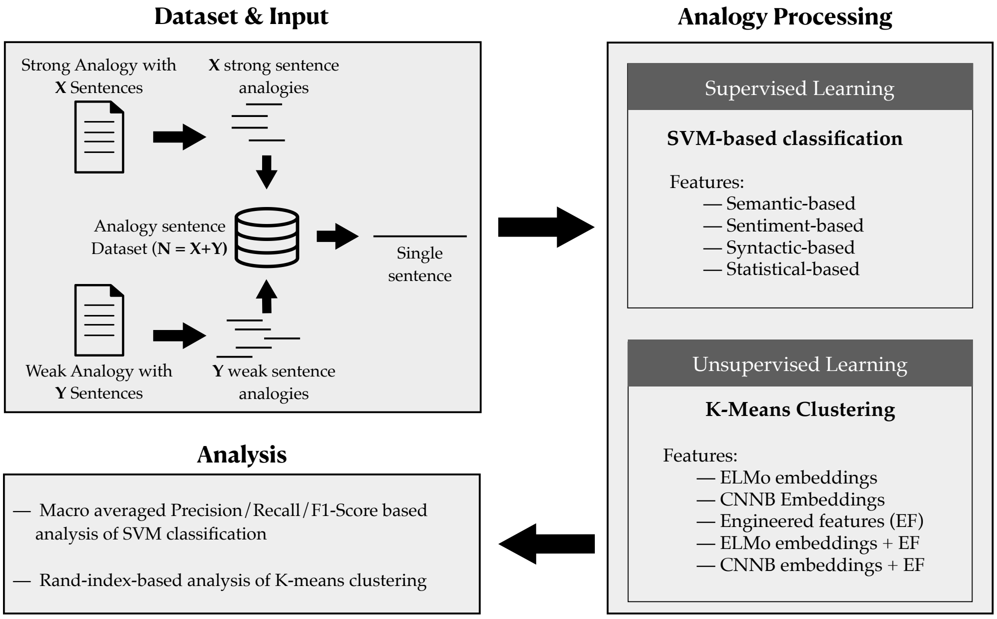


Figure 1: Illustration of the DAF Pipeline. The analogies are split into sentences to create the analogy dataset. Single sentences are sent through Analogy Processing. Features of the sentences are extracted and used in SVM-based classification and K-means clustering. An analysis is then conducted on SVM-based classification and K-means clustering results.

2.1. Dataset and Input

The dataset used in this work was drawn from 500 student-created analogies, collected in a college classroom. An instructor explained a process in the domain of biochemistry, e.g., Glycolysis (source analogy), and requested students to construct a scenario analogous to the explained process from a domain of their choice (target analogy). The instructor then evaluated 31 student-generated analogous scenarios as a strong or weak analogy based on its correspondence to the Biochemistry concept. A strong analogy corresponds well with the target analogy, and a weak analogy minimally corresponds with the target analogy. To increase the size of the 31 exemplar data set from the original we split each analogy into its constituent sentences, generating a data set of 526 strong exemplars and 140 weak exemplars. Each constituent sentence of an analogy falls into the same annotation category as the original analogy. *Ergo, the initial input to the DAF pipeline is a sentence.* This work does not distinguish between the analogy's target domains (Enzyme Kinetics and Glycolysis). Table 1 presents the summarized statistics of the dataset.

Table 1
Dataset statistics

	Strong	Weak
Num. of analogies	25	6
Num. of analogies (sentences)	586	140

2.2. Input Processing

Sentences were processed and used as inputs to a Support Vector Machine (SVM) classifier (supervised learning) and K-means clustering (unsupervised learning) separately. In the following section we briefly review the background of input processing techniques, learning methods and implementation details.

2.2.1. Background

SVM is a supervised learning technique that creates functions to map inputs to pre-existing annotations [14]. SVM is an easy-to-interpret classifier providing competitive performance in classification, regression, and outlier detection tasks [15]. The following paragraphs detail the background of four feature groups of interest here.

The obviously relevant features are semantic. Abstract Meaning Representation (AMR) is a semantic representation language that expresses a sentence's logical meaning by converting it to a rooted, directed, acyclic, edge-labeled, and leaf-labeled graph. [16]. To abstract away from syntactic idiosyncrasies, AMR assigns the same AMR graph to sentences with the same *meaning*. Nodes of an AMR graph are labeled as *concepts*, edges as *relations*, and concept properties as attributes. Concepts are either English words, PropBank framesets [17] or special keywords. There are approximately 100 relations [16]. AMR is used as a semantic representation of text in several NLP tasks such as summarization [18], machine comprehension [19], and event extraction [20, 21]. In this work we use AMR representations to extract concepts, relations and attributes present in sentence analogies. Figure 2 illustrates the AMR for a sentence from the dataset.

Sentiment-based features potentially reveal student engagement. Sentiment analysis aims to identify emotional or affective tendencies in user-generated content such as tweets, product reviews, and feedback [22]. Subjectivity detection and polarity determination are two common tasks in sentiment analysis [23]. Subjectivity quantifies the personal opinions versus factual information contained in the text. High subjectivity indicates the text contains more personal opinions compared to factual information [23]. Polarity describes the sentiment of a piece of text as positive, negative, or neutral [22].

We extract three groups of syntactic features. The first feature group is Part of Speech (POS), a grammatical classification of the word types in a sentence. These POS tags commonly include nouns, verbs, adjectives, etc. [24]. Named Entities Recognition (NER), the second feature group, is used to identify occurrences of named entities such as people, organizations, times, and locations in a sentence [25]. The third feature group is sentence type. Sentences in the dataset are identified as complex or compound sentences and simple sentences. In linguistics, complex sentences are sentences with two or more clauses connected with a subordinate conjunction. Simple sentences contain one independent clause [26].

We use four routine and straightforward statistical features, word count, character count, the average word length of a sentence (character count/ word count), and the number of unique words in a sentence.

K-means is a non-deterministic, iterative, and unsupervised machine learning technique to produce clusters from data [27]. Unsupervised learning here serves as both a baseline for

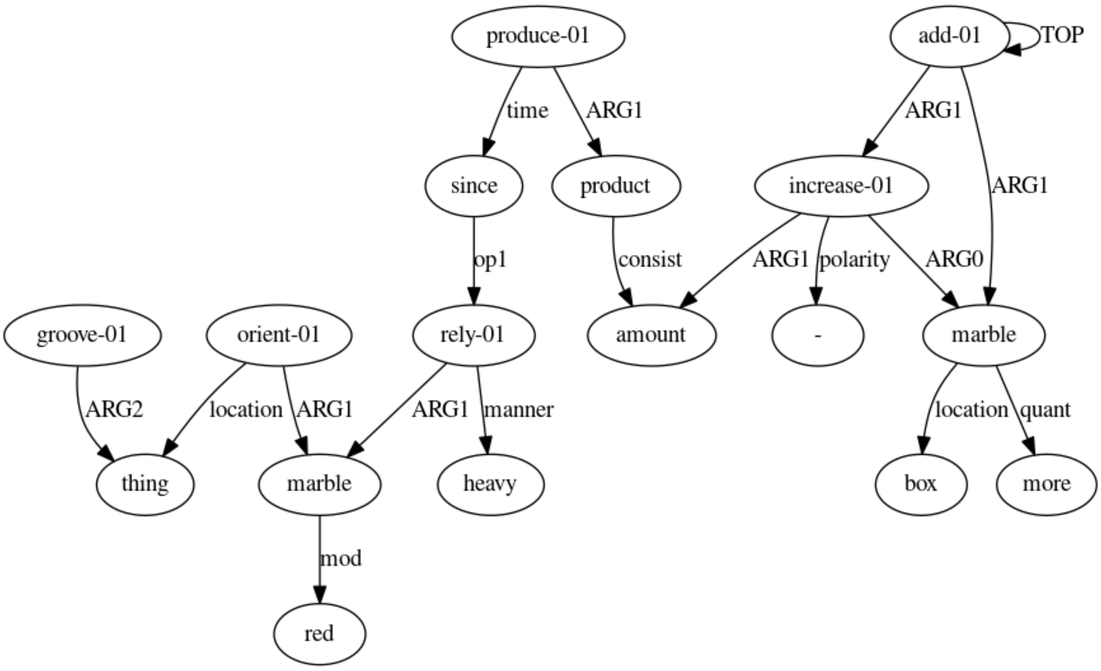


Figure 2: AMR representation of the sentence *"Adding more marbles to the box will not increase the amount of product produced since it relies heavily on red marbles being oriented into the grooves."* from the dataset.

comparison with supervised learning results, and as a long term goal in itself, independent of any manual annotation. In the simplest test, we converted the input sentences to embeddings and cluster them using K-means clustering. Sentence embeddings were created using two techniques, context-based Embeddings from Language Models (ELMo) and knowledge-graph-based ConceptNet Numberbatch (CNNB). The following two paragraphs give a brief overview of these two embedding techniques.

ELMo embeddings are deep, contextualized representations of words computed using a two-layer bidirectional language model (biLM), which is pretrained on a large text corpus [28]. ELMo is robust in creating embeddings for out-of-vocabulary (OOV) words because it incorporates subword and character-level information when creating embeddings. Handling OOV words is particularly important in this work as most of the sentences often contain domain-specific keywords such as *"Glucose-6-p", "DHP", "GAP"* which can fall into the OOV category. ELMo sentence vectors are 1024-dimensional.

ConceptNet is a semantic network of knowledge about word meanings [29]. CNNB embeddings [30] are semantic word vectors created by encoding the knowledge in ConceptNet [29]. ConceptNet Numberbatch sentence embeddings are produced by taking the mean of single word embeddings in a sentence. CNNB sentence vectors are 300-dimensional. CNNB embeddings are not as robust as ELMo embeddings when handling OOV words, yet the percentage of OOV words in the current dataset is rather small (6%). Hence we use CNNB as the second embedding technique to create sentence embeddings.

2.2.2. Implementation Details

We use Pandas DataFrames [31] to process and manipulate the sentence features. We also used other external libraries used in the extraction of sentence features as follows. To extract semantic features, the sentences are sent through a transition-based AMR parser named CAMR [32]. Textblob [33] is used to assess the subjectivity and polarity scores of the sentences. POS tag and NER-related features (in syntactic features category) are extracted using spaCy [1]. Matplotlib [2] and seaborn [3] are used for the visualizations.

ConceptNet Numberbatch embeddings are static representations for words available publicly [4]. ELMo sentence embeddings were created using the model available at Tensorflow Hub[5].

2.3. Analysis

In the following sections, we look at semantic, sentiment-based, syntactic, and statistical feature distributions for strong and weak analogies. We then compare the performances of an SVM classifier and K-means clustering.

Figure 3 illustrates the distributions of counts of concepts, relations, attributes, unique concepts, unique relations, and unique attributes of strong and weak analogies. Figure 4 presents the polarity and subjectivity distribution of strong and weak analogies. As shown in the plots, both strong and weak analogies contain sentences with neutral polarity and less subjectivity. Seventeen POS tags are present in the dataset. Distributions of the three most prevalent POS tags in strong and weak analogies are depicted in Figure 5 to utilize space effectively. Nevertheless, we used all 17 POS tags in the SVM classifier as features. Out of the fourteen named entities in the dataset (that are used in the SVM classifier), the distributions of the top three (ORG, CARDINAL, and PERSON) are plotted in Figure 6. We use the spaCy English pipeline [6] for NER tagging. Analogies written by students contain several references to biochemicals. These are misidentified as organizations (ORG) by spaCy, resulting in the ORG tag being the top named entity in the dataset. Figure 7 shows the distribution of simple and complex/ compound sentences. Weak analogies tend to have a slightly higher number of complex/ compound sentences, and strong analogies have slightly more simple sentences. Figure 8 presents the distributions of word counts, character counts, average word lengths, and unique word counts of strong and weak analogies. Modest discrepancies between distributions suggest the potential for such features to distinguish between strong and weak analogies. Therefore a feature vector combining the abovementioned features (engineered features) was then used in

[1] https://spacy.io/
[2] https://matplotlib.org/
[3] https://seaborn.pydata.org/
[4] https://github.com/commonsense/conceptnet-numberbatch
[5] https://tfhub.dev/google/elmo/3
[6] https://spacy.io/models/en#en_core_web_md

an SVM classifier to classify strong and weak analogies.
We use five variants of sentence vectors as inputs to the SVM classifier and K-means clustering. The first variant is the ELMo embeddings vector (ELMo). The second variant is the CNNB embeddings vector (CNNB), and the third variant is the engineered features vector with the vector dimension of 44. The fourth variant is a simple concatenation between ELMo embeddings and the engineered feature vector (ELMo composite). The fifth variant is a simple concatenation between CNNB embeddings and the engineered feature vector (CNNB composite).

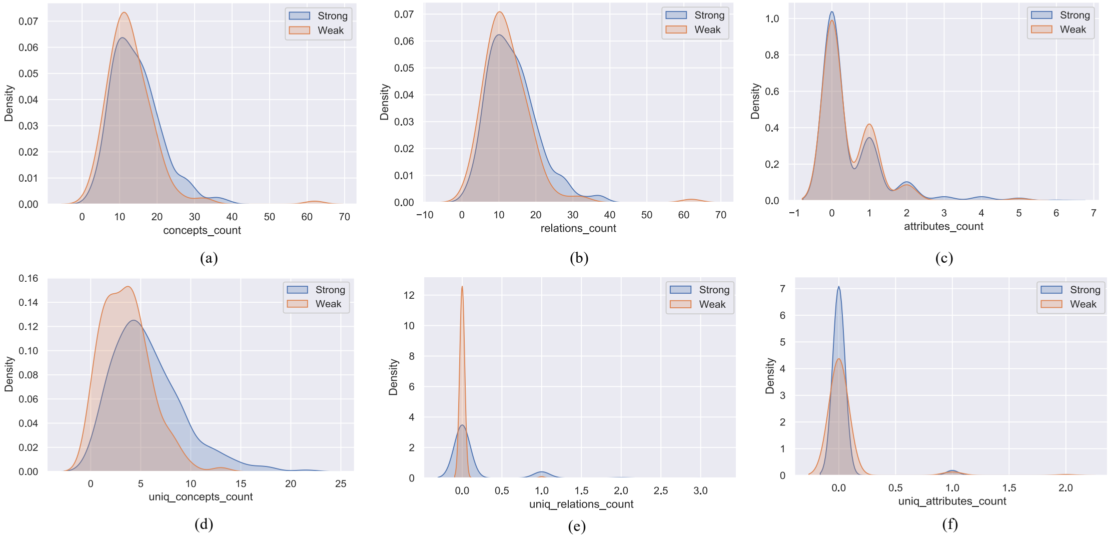


Figure 3: Plots illustrating the distributions of (a) Concepts counts, (b) Relations counts, (c) Attributes counts, (d) Unique concepts counts, (e) Unique relations counts, and (f) Unique attributes counts of sentence analogies.

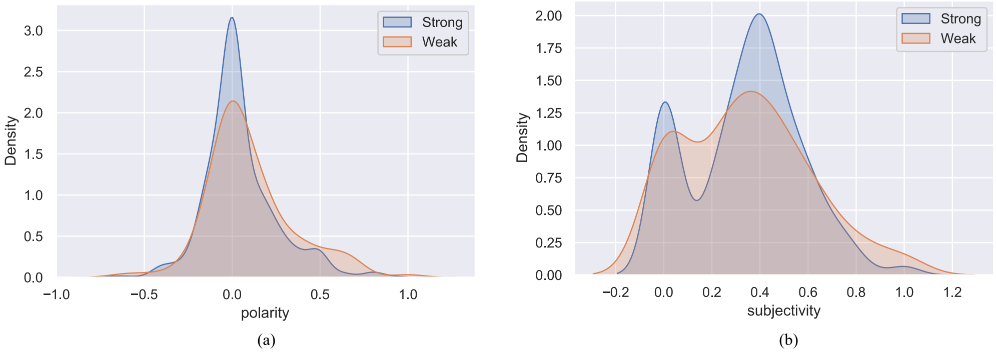
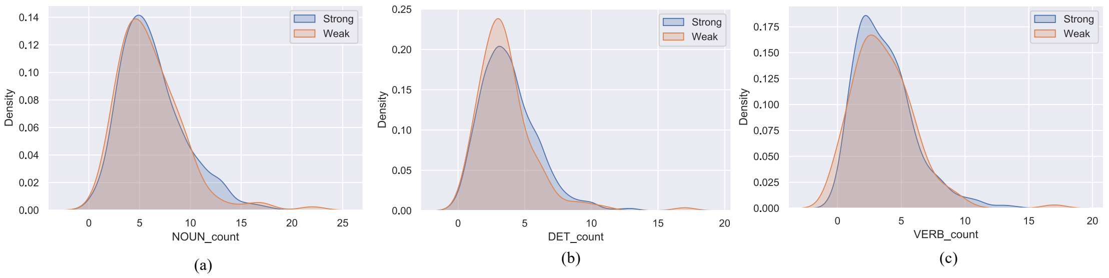


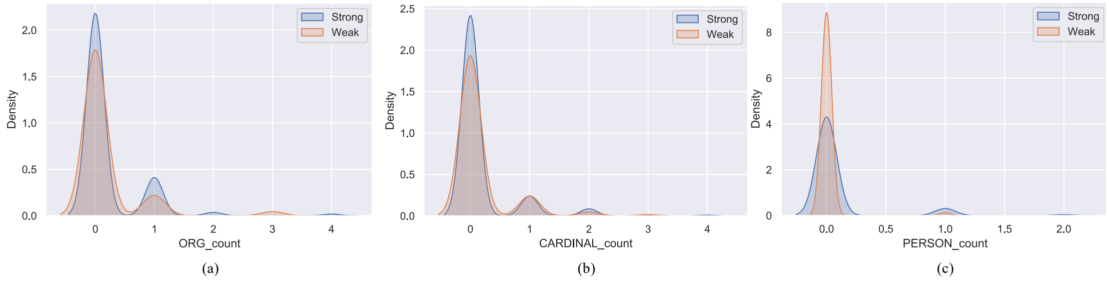
Figure 4: Plots illustrating the distributions of (a) Polarity, (b) Subjectivity across sentence analogies.

We opted to train an SVM classifier with stratified K-fold cross validation due to the limited size of our dataset (less than 1000 data points). Due to the imbalanced nature of the dataset and



(a) (b) (c)

Figure 5: (a) VERB, (b) DETERMINER, (c) NOUN POS tags counts distribution across sentence analogies.



(a) (b) (c)

Figure 6: (a) ORG, (b) CARDINAL, and (c) PERSON NER tag counts distribution across sentence analogies.

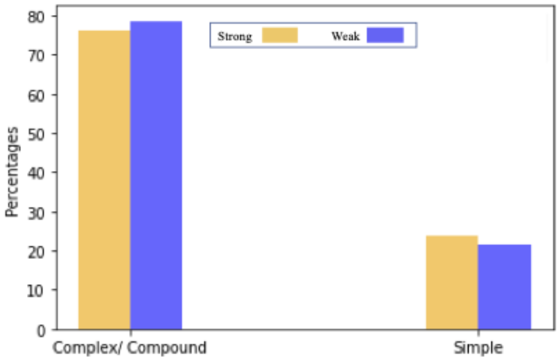


Figure 7: Sentence type statistics of strong and weak analogies.

identifying strong and weak analogies were equally important in this initial analysis, we used macro-F1 as the performance metric [34]. Performance of the SVM classifier with five variants of sentence vectors are listed in table 2.3.

We further inspect the contributions of the engineered features from the four feature categories mentioned in section 2.2 when discriminating between strong and weak analogies. We observe (see Figure 9) that most of the top 15 discriminating features belong to the syntactic

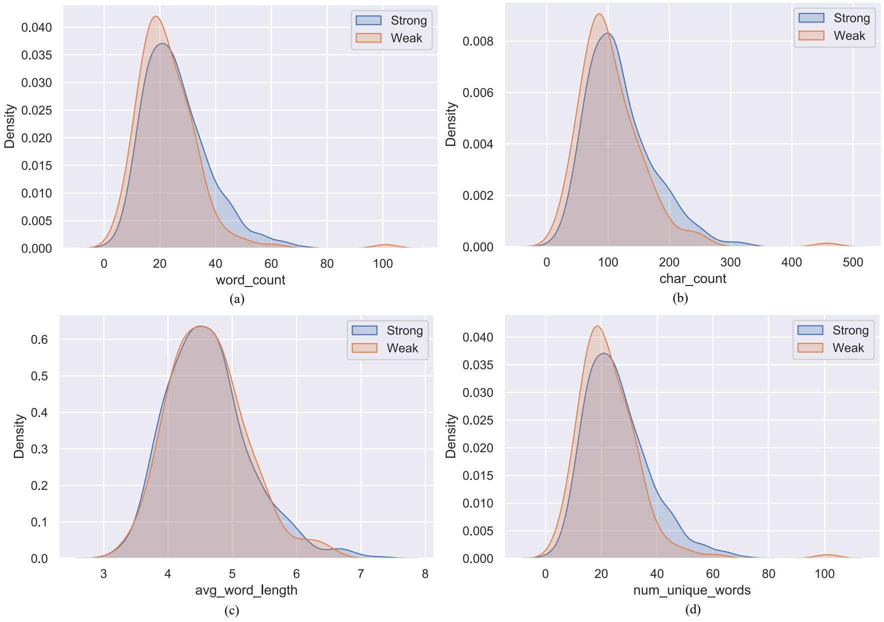


Figure 8: Plots illustrating the distributions of (a) Word count, (b) Character count, (c) Avg. word length, and (d) Number of unique words across sentence analogies.

feature category, but a semantic feature contributes the most to discriminate between strong and weak analogies.

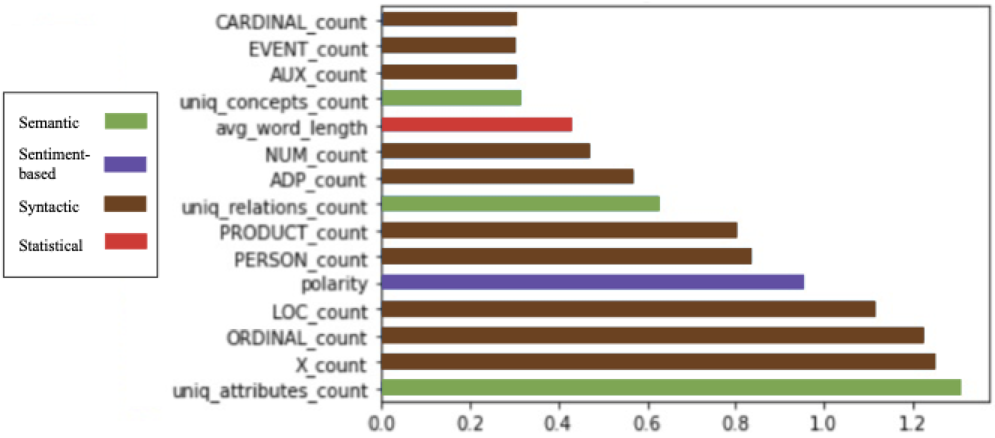


Figure 9: Top 15 discriminative features between strong and weak analogies

We use K-means to cluster the five variants of sentence vectors mentioned above with cluster centers randomly selected and K set to two. Based on the Rand index [7], the clusters are not well

[7] https://scikit-learn.org/stable/modules/clustering.html#rand-index

Table 2
Different sentence vector variants and their performance on SVM classifier measured by *macro* averaged Precision/Recall/F1-score along with their K-means cluster qualities given by Rand Index.

Vector Variant	SVM Classifier			K-Means Clustering
	Precision	Recall	F1-score	Rand Index
ELMo embeddings	0.96	0.91	0.93	0.51
CNNB embeddings	0.86	0.82	0.84	0.52
Engineered features	0.75	0.64	0.67	0.54
ELMo composite	0.96	0.96	0.96	0.50
CNNB composite	0.83	0.82	0.81	0.52

segregated in any variant, yet engineered features embeddings performed slightly better (see Table 2.3).

3. Discussion

This section presents our findings and insights, followed by the limitations and future work.

3.1. Findings and Insights

We introduce the DAF pipeline to identify discriminative features of strong and weak analogies. We show that just a few engineered features does a surprisingly good job as input to SVM. To be sure, the ELMo composite sent through the SVM classifier performs better than the rest of the sentence vector variants. Nevertheless, the ELMo composite score is slightly higher (∼0.03) than the ELMo. This increase highlights that the engineered features encode some aspects of the analogies not well-captured by the ELMo embeddings. Although the SVM's performance with the engineered feature vector is 26% lower than that of the ELMo embedding, its embedding size is ∼23 times smaller than the ELMo. This phenomenon hints that considerable performance gains can be achieved with a much smaller number of better hand-crafted features, and most importantly, the better performance is explainable. We also note that the CNNB composite vector's performance in SVM is slightly poorer than that of the CNNB itself (∼0.02). Although further exploration is required to explain this phenomenon clearly, we suspect this may be the result of feature multicollinearity specific to the manner in which CNNB creates its embeddings, combined with idiosyncracies of the subsets constructed in cross-validation.

We show that, among the features passed to the SVM classifier, the most discriminative feature for classification is a semantic feature (unique attribute count) and three out of the four semantic features (unique relations count, unique concepts count, concepts count) fall in the list of top 15 discriminative features. Also, among the top 15 discriminative features, syntactic features have the most representation. Overall, the engineered features are few in number, meaningful, and relatively cheap to calculate. Given the range of content in the data set –anything from marbles to cake–the modest success reported here is impressive. These features will contribute to our future efforts based on more computationally intensive semantic analysis. A successful unsupervised learning method would liberate classifier training from dependence on manual annotation. Unsupervised learning results remain largely unimpressive. Nevertheless, there are some hints of promise. Engineered features improve clustering results relative to embeddings alone or embeddings and engineered features. This reinforces our claim that such features are identifying discriminators that are not captured by embeddings.

3.2. Limitations and Future Work.

The dataset used in this work is imbalanced, with more strong analogy data points than weak ones. This may cause the "uniform effect" where K-means produces clusters of the same size, even when the "true" cluster sizes of the dataset are varied [35]. To overcome such issues we plan to improve class imbalance through SMOTE [36], GANS[37], and the expansion of the manually-annotated corpus.

The natural language processing techniques employed in this work do not handle the particular nature of the dataset. For example, the spaCy model we use is trained on a generic English corpus [8]. However, we plan to use models/ techniques trained on subject-specific corpora to overcome issues like misidentifying biochemical terms as organizations in NER. Also, students use the term "like" in their target analogies to signify the similarity between their analogy and the source domain (biochemistry) concept. These are wrongly picked up by the sentiment analysis tool when evaluating polarity. Modified corpora will allow us to better manage these issues.

We classified analogy strength using individual sentences, which is both a benefit and a limitation. As a result, we identified very simple discriminators. However, some sentences in the dataset might not contribute when creating strong/ weak analogies. Constraining analysis to the sentence level requires annotation to eliminate this potential source of noise. However, the long-term goal is to evaluate analogies at the document level, for their epistemic quality. Though still vector based, our ongoing work in this area employs referent knowledge bases for both the target and variable student sources, to guide semantic interpretation.

4. Conclusion

This work introduces the DAF pipeline to identify discriminative features between strong and weak long-form analogies. We show that an SVM-based supervised-learning approach can successfully discriminate component sentences drawn from strong and weak analogies. Semantic and several syntactic features are the main contributors to discrimination, helping us to realize our goal of efficient evaluation of student generated long-form analogies.

Acknowledgments

We thank Dr. Biplav Srivastava for his valuable feedback and Dr. Nitin Jain for providing the data used in this work. We also thank the reviewers for their constructive comments.

[8] https://spacy.io/models/en#en_core_web_md

References

[1] D. Gentner, Structure-mapping: A theoretical framework for analogy, Cognitive science 7 (1983) 155–170.

[2] D. J. Chalmers, R. M. French, D. R. Hofstadter, High-level perception, representation, and analogy: A critique of artificial intelligence methodology, Journal of Experimental & Theoretical Artificial Intelligence 4 (1992) 185–211.

[3] K. J. Holyoak, P. Thagard, Mental leaps: Analogy in creative thought, MIT press, 1996.

[4] D. C. Krawczyk, R. G. Morrison, I. Viskontas, K. J. Holyoak, T. W. Chow, M. F. Mendez, B. L. Miller, B. J. Knowlton, Distraction during relational reasoning: The role of prefrontal cortex in interference control, Neuropsychologia 46 (2008) 2020–2032.

[5] R. G. Morrison, D. C. Krawczyk, K. J. Holyoak, J. E. Hummel, T. W. Chow, B. L. Miller, B. J. Knowlton, A neurocomputational model of analogical reasoning and its breakdown in frontotemporal lobar degeneration, Journal of cognitive neuroscience 16 (2004) 260–271.

[6] N. Ichien, H. Lu, K. J. Holyoak, Verbal analogy problem sets: An inventory of testing materials, Behavior research methods 52 (2020) 1803–1816.

[7] H. Prade, G. Richard, Analogical proportions: Why they are useful in ai., in: IJCAI, 2021, pp. 4568–4576.

[8] S. Lim, H. Prade, G. Richard, Classifying and completing word analogies by machine learning, International Journal of Approximate Reasoning 132 (2021) 1–25.

[9] A. Ushio, L. Espinosa-Anke, S. Schockaert, J. Camacho-Collados, BERT is to NLP what alexnet is to CV: Can pre-trained language models identify analogies?, in: ACL 2022 Workshop on Commonsense Representation and Reasoning, 2022. URL: https://openreview.net/forum?id=BdWgrMFxdW9.

[10] B. A. Spellman, K. J. Holyoak, Pragmatics in analogical mapping, Cognitive psychology 31 (1996) 307–346.

[11] B. Falkenhainer, K. D. Forbus, D. Gentner, The structure-mapping engine: Algorithm and examples, Artificial intelligence 41 (1989) 1–63.

[12] M. McLure, S. Friedman, K. Forbus, Extending analogical generalization with near-misses, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 29, 2015.

[13] R.Shrem, T. Vonderhaar, V. Shalin, N. Jain, Use of student-generated process analogies to enhance student engagement, in: (in preparation), 2022.

[14] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297.

[15] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, A. Lopez, A comprehensive survey on support vector machine classification: Applications, challenges and trends, Neurocomputing 408 (2020) 189–215.

[16] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, N. Schneider, Abstract Meaning Representation for sembanking, in: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 178–186. URL: https://aclanthology.org/W13-2322.

[17] P. Kingsbury, M. Palmer, From TreeBank to PropBank, in: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02), European Language Resources Association (ELRA), Las Palmas, Canary Islands - Spain, 2002. URL:

http://www.lrec-conf.org/proceedings/lrec2002/pdf/283.pdf.

[18] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, N. A. Smith, Toward abstractive summarization using semantic representations, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 1077–1086. URL: https://aclanthology.org/N15-1114. doi:10.3115/v1/N15-1114.

[19] M. Sachan, E. Xing, Machine comprehension using rich semantic representations, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2016, pp. 486–492.

[20] S. Rao, D. Marcu, K. Knight, H. Daumé III, Biomedical event extraction using Abstract Meaning Representation, in: BioNLP 2017, Association for Computational Linguistics, Vancouver, Canada„ 2017, pp. 126–135. URL: https://aclanthology.org/W17-2315. doi:10.18653/v1/W17-2315.

[21] L. Huang, T. Cassidy, X. Feng, H. Ji, C. Voss, J. Han, A. Sil, Liberal event extraction and event schema induction, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 258–268.

[22] C. Puschmann, A. Powell, Turning words into consumer preferences: How sentiment analysis is framed in research and the news media, Social Media+ Society 4 (2018) 2056305118797724.

[23] E. Kasmuri, H. Basiron, Subjectivity analysis in opinion mining—a systematic literature review, Int J Adv Soft Comput Appl 9 (2017) 132–159.

[24] A. Chiche, B. Yitagesu, Part of speech tagging: a systematic review of deep learning and machine learning approaches, Journal of Big Data 9 (2022) 1–25.

[25] A. Mikheev, M. Moens, C. Grover, Named entity recognition without gazetteers, in: Ninth Conference of the European Chapter of the Association for Computational Linguistics, 1999, pp. 1–8.

[26] B. Das, M. Majumder, S. Phadikar, A novel system for generating simple sentences from complex and compound sentences, International Journal of Modern Education and Computer Science 11 (2018) 57.

[27] J. MacQueen, Classification and analysis of multivariate observations, in: 5th Berkeley Symp. Math. Statist. Probability, 1967, pp. 281–297.

[28] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237. URL: https://aclanthology.org/N18-1202. doi:10.18653/v1/N18-1202.

[29] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: Thirty-first AAAI conference on artificial intelligence, 2017.

[30] R. Speer, J. Chin, An ensemble method to produce high-quality word embeddings, arXiv preprint arXiv:1604.01692 (2016).

[31] W. McKinney, et al., pandas: a foundational python library for data analysis and statistics, Python for high performance and scientific computing 14 (2011) 1–9.

[32] C. Wang, S. Pradhan, X. Pan, H. Ji, N. Xue, Camr at semeval-2016 task 8: An extended

transition-based amr parser, in: Proceedings of the 10th international workshop on semantic evaluation (semeval-2016), 2016, pp. 1173–1178.

[33] S. Loria, et al., textblob documentation, Release 0.15 2 (2018).

[34] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Information processing & management 45 (2009) 427–437.

[35] H. Xiong, J. Wu, J. Chen, K-means clustering versus validation measures: A data-distribution perspective, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39 (2009) 318–331. doi:10.1109/TSMCB.2008.2004559.

[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).