Accepted Manuscript online: 2022-05-25 Article published online: 2022-09-19

A Systematic Approach to Configuring MetaMap for Optimal Performance

Xia Jing[1] Akash Indani[2] Nina Hubig[2] Hua Min[3] Yang Gong[4] James J. Cimino[5] Dean F. Sittig[4] Lior Rennert[1] David Robinson[6] Paul Biondich[7] Adam Wright[8] Christian Nøhr[9] Timothy Law[10] Arild Faxvaag[11] Ronald Gimbel[1]

[1] Department of Public Health Sciences, College of Behavioral, Social and Health Sciences, Clemson University, Clemson, South Carolina, United States
[2] School of Computing, College of Engineering, Computing and Applied Sciences, Clemson University, Clemson, South Carolina, United States
[3] Department of Health Administration and Policy, College of Health and Human Services, George Mason University, Fairfax, Virginia, United States
[4] School of Biomedical Informatics, The University of Texas Health Sciences Center at Houston, Houston, Texas, United States
[5] Informatics Institute, The University of Alabama at Birmingham, Birmingham, Alabama, United States
[6] Independent Consultant, Cumbria, United kingdom
[7] Department of Pediatrics, Clem McDonald Biomedical Informatics Center, Regenstrief Institute, Indiana University School of Medicine, Indianapolis, Indiana, United States
[8] Department of Biomedical Informatics, Vanderbilt University Medical Center, Nashville, Tennessee, United States
[9] Department of Planning, Faculty of Engineering, Aalborg University, Aalborg, Denmark
[10] Ohio Musculoskeletal and Neurologic Institute, Ohio University, Athens, Ohio, United States
[11] Department of Neuromedicine and Movement Science, Faculty of Medicine and Health Sciences, Norwegian University of Science and Technology, Trondheim, Norway

Address for correspondence Xia Jing, MD, PhD, Department of Public Health Sciences, College of Behavioral, Social and Health Sciences, Clemson University, Edwards Hall 511, Clemson, SC 29634, United States (e-mail: xjing@clemson.edu).

Methods Inf Med 2022;61:e51–e63.

Abstract

Background MetaMap is a valuable tool for processing biomedical texts to identify concepts. Although MetaMap is highly configurative, configuration decisions are not straightforward.

Objective To develop a systematic, data-driven methodology for configuring MetaMap for optimal performance.

Methods MetaMap, the word2vec model, and the phrase model were used to build a pipeline. For unsupervised training, the phrase and word2vec models used abstracts related to clinical decision support as input. During testing, MetaMap was configured with the default option, one behavior option, and two behavior options. For each configuration, cosine and soft cosine similarity scores between identified entities and

Keywords

- MetaMap
- natural language processing
- clinical decision support system
- configuration and optimization
- performance

article published online
September 6, 2022

DOI https://doi.org/10.1055/a-1862-0421.
ISSN 0026-1270.

Georg Thieme Verlag KG, Rüdigerstraße 14, 70469 Stuttgart, Germany

gold-standard terms were computed for 40 annotated abstracts (422 sentences). The similarity scores were used to calculate and compare the overall percentages of exact matches, similar matches, and missing gold-standard terms among the abstracts for each configuration. The results were manually spot-checked. The precision, recall, and F-measure ($\beta = 1$) were calculated.

Results The percentages of exact matches and missing gold-standard terms were 0.6–0.79 and 0.09–0.3 for one behavior option, and 0.56–0.8 and 0.09–0.3 for two behavior options, respectively. The percentages of exact matches and missing terms for soft cosine similarity scores exceeded those for cosine similarity scores. The average precision, recall, and F-measure were 0.59, 0.82, and 0.68 for exact matches, and 1.00, 0.53, and 0.69 for missing terms, respectively.

Conclusion We demonstrated a systematic approach that provides objective and accurate evidence guiding MetaMap configurations for optimizing performance. Combining objective evidence and the current practice of using principles, experience, and intuitions outperforms a single strategy in MetaMap configurations. Our methodology, reference codes, measurements, results, and workflow are valuable references for optimizing and configuring MetaMap.

Introduction

Natural language processing (NLP) is an important component of artificial intelligence and is critical for a computer-based understanding of human languages.[1,2] In the biomedical and health fields, NLP is used to discover new disease risk factors,[3] detect or predict significant clinical events from existing texts (e.g., patient records, clinical notes), and summarize texts automatically to facilitate clinical documentation, especially via electronic health records.[4,5] Identifying computer-processable concepts from narrative texts is a critical first task in understanding the natural language.

MetaMap[6,7] is a valuable tool in the biomedical and health NLP fields. Over the past several decades, MetaMap has been widely used to facilitate indexing, data mining, and other NLP projects.[8–11] To identify computer-processable concepts in narrative texts, MetaMap leverages the Unified Medical Language System (UMLS).[12] Both MetaMap and UMLS were developed and are maintained by the National Library of Medicine. A component of UMLS, Metathesaurus, includes most vocabularies and coding standards in the biomedical and health fields. Thus, UMLS is a cornerstone that enables the interoperability of health information systems.[13,14]

MetaMap comprehensively covers biomedical subjects and is highly configurable. Several parameters can be configured, including vocabulary sources drawn from Metathesaurus, options related to semantic types of concepts in Metathesaurus, data (e.g., UMLS version, Data Model), output/display (e.g., Display Variants, Show Candidates), browse mode (e.g., Allow Overmatch), and behavior. For behavior, 17 options (e.g., Enable NegEx, Use Word Sense Disambiguation) can be configured. Users usually select specific vocabulary sources and semantic types based on the biomedical domain of the texts and the desired annotation targets when configuring MetaMap. Selecting other options is less straightforward, as they are related to NLP techniques rather than biomedical concepts.

Researchers have been developing multipurpose tools based on MetaMap.[15,16] For example, Pratt and colleague compared the performance between MetaMap and humans to identify concepts from the titles of articles as a form of narrative texts.[17] However, no one has yet *published a systematic comparison* of MetaMap performance based on different configurations. In addition to relying on the current and limited process of configuring MetaMap based on principles, intuitions, and experience, we also need objective guidance for configuring MetaMap to optimize its performance.

We are constructing ontology for characterizing a clinical decision support system (CDSS) and using MetaMap and other tools to process the published literature on CDSS to identify candidate concepts. The ontology intends to generate CDSS rules to improve the interoperability of the CDSS rules. Through this work, we realized that optimally configuring MetaMap is critical and necessary but challenging. Although we reviewed the current literature on MetaMap, studies describing formal methodologies for configuring MetaMap were not found. Therefore, we examined how different configurations affect MetaMap performance. This article describes our systematic approach and the results of applying our methodology to configure MetaMap for optimal performance.

Objectives

This study was performed to explore a systematic, data-driven methodology for configuring MetaMap more accurately with robust evidence for optimal performance.

Methods

Workflow and Experimental Design

We followed two main steps in this experiment: training and testing (►Fig. 1). We first extracted 3,187 journal article abstracts on CDSS from PubMed (the search strategies are described in ►Supplementary Appendix A). We then implemented preprocessing steps that included the following: (1) removing extra whitespace; (2) removing stop words (e.g., the, an, is, and); (3) removing special characters (e.g., "/!#@ $-); and (4) removing numeric values. Punctuations were retained during the preprocessing as MetaMap uses them to divide the text into phrases. After preprocessing, the entire set of abstracts was used to train (unsupervised) both the phrase model and the word2vec model to obtain the maximum vocabulary and original contexts. Training the phrase model generates phrases with bigrams (more than one word) without distinguishing the importance of particular phrases. The word2vec model,[18,19] a two-layer neural network, is used to (1) compare identified entities with gold-standard terms in our areas of focus (i.e., CDSS) during testing and (2) convert words into continuous vectors during training and testing. Here, we describe part of a pipeline being developed to identify entities from abstracts.

From the 3,187 abstracts, we randomly selected 44 abstracts for the testing step. Three annotators with a medical background annotated these abstracts to create *gold standards*. We then added MetaMap, with different configurations, to the pipeline to process the annotated abstracts before entering the trained phrase model and word2vec model (►Fig. 1). MetaMap was used to identify concepts from the annotated abstracts. The identified concepts could be nouns, verbs, a single word, or a phrase. These concepts were used to form phrases (e.g., bigrams or trigrams) via the trained phrase model. The phrase model's output was then converted into vectors using the trained word2vec model. The goals of the testing phase were to (1) understand the basic performance of the current pipeline in processing all abstracts during training and (2) compare different configurations of MetaMap using multiple measurements.

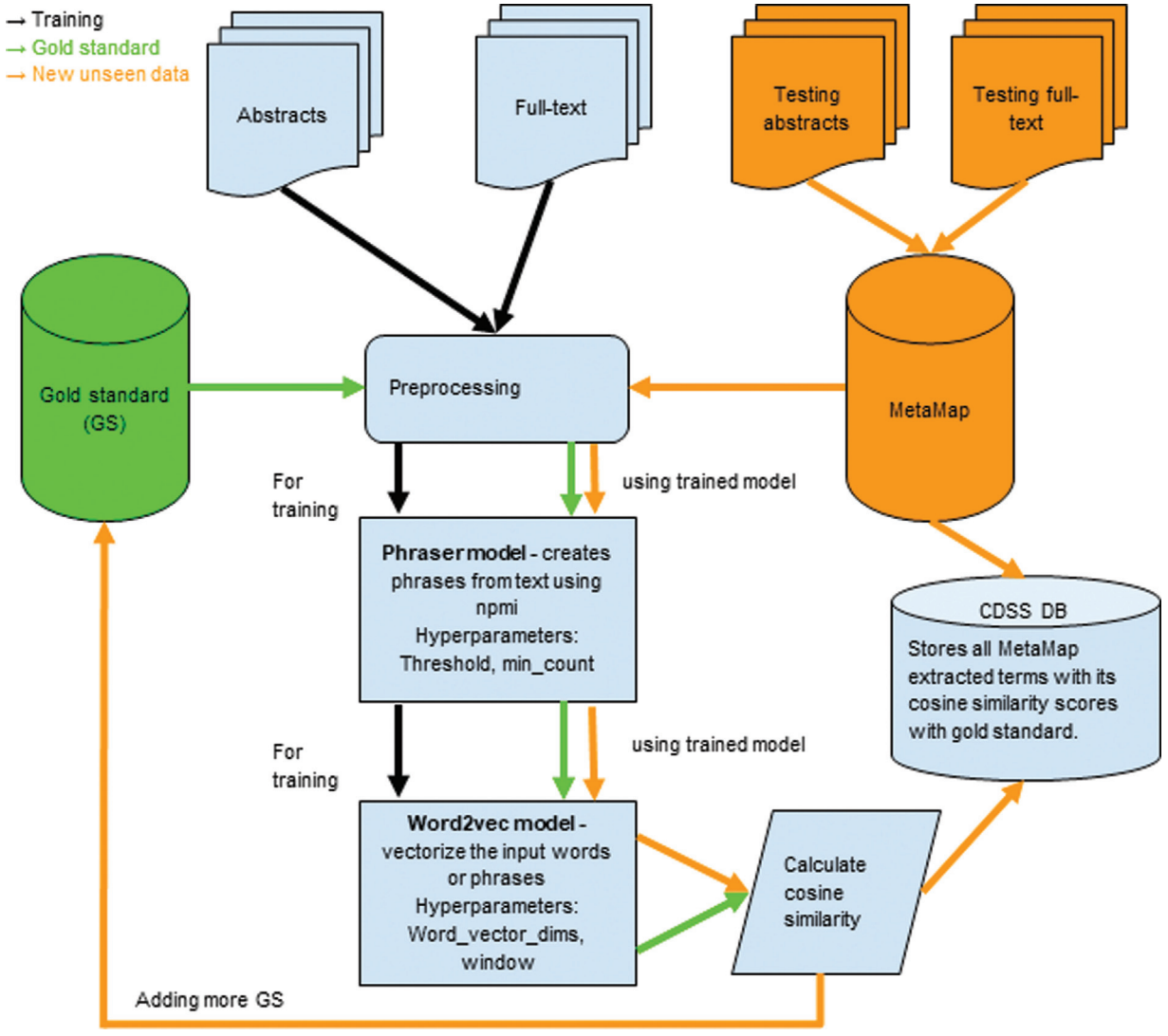

Fig. 1 Overall experimental flow of training and testing. *Blue blocks* and *dark arrows* show the flow of training. *Orange blocks* and *arrows* indicate the flow of testing. The *green block* and *arrows* show the flow of gold standards used only during testing. CDSS, clinical decision support system; DB, database; npmi, normalized pointwise mutual information.

For each identified term, we computed cosine and soft cosine similarity scores based on the vector of this term and the vectors of the three best-matched gold-standard terms.[20,21] The cosine and soft cosine similarity scores were used to indicate the similarity between these vectors. We then categorized the numeric similarity scores into exact matches ($\geq$0.85), similar matches ($\geq$0.65 and $<$0.85), and missing gold-standard terms ($<$0.65).

Of the 44 annotated abstracts, 40 were processed for each configuration to identify entities (single words, bigrams, trigrams), and vectors for each identified entity were obtained. Within each abstract, similarity scores were calculated for each identified entity. Based on the similarity scores, we calculated the percentages of exact matches, similar matches, and missing gold-standard terms. Finally, we averaged the percentages of exact matches, similar matches, and missing gold-standard terms among the annotated 40 abstracts as performance indicators for each MetaMap configuration (►Fig. 2). The remaining four annotated abstracts were used in a pilot study to test whether the pipeline was operational before the experiments.

For each computation, we set different MetaMap configurations to default (no option was selected), one behavior option (►Table 1–[22,23]), and two behavior options with reasonable combinations (►Table 2). By "reasonable combinations," we mean that we would not select "prune threshold" and "disable pruning" as a combination during permutation and combination. MetaMap can have 12 distinct configurations for one behavior option and 63 for two behavior options. Each configuration is measured by percentages of exact matches, similar matches, and missing gold-standard terms of all 40 abstracts. For each abstract, percentages of exact matches, similar matches, and missing gold-standard terms were calculated based on the cosine and soft cosine similarity scores between each identified entity and gold-standard terms.

We examined the validity of the similarity scores generated from the current pipeline with a manual spot-check. We randomly selected 8 of the 40 annotated abstracts. Two were assigned to each of the following categories: cosine with one behavior option, soft cosine with one behavior option, cosine with two behavior options, and soft cosine with two behavior options. We selected the highest and lowest percentages for the exact matches for each abstract within every category. The average precision, recall, and F-measure ($\beta = 1$) were calculated for the exact matches, similar matches, and missing gold-standard terms. ►Supplementary Appendix B (describes the principles used to determine matches during the manual spot-check.

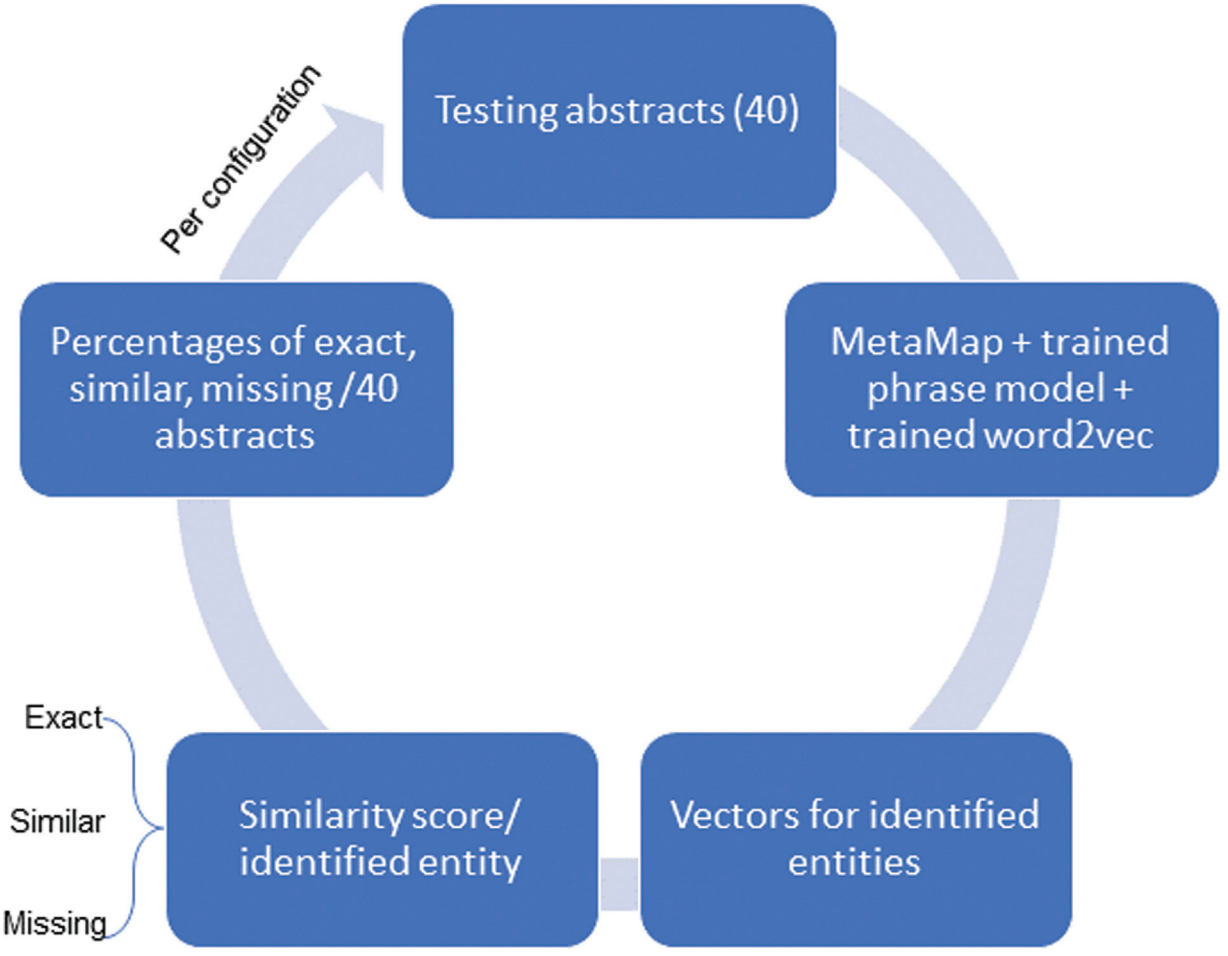

Fig. 2 Process and calculation flow for each MetaMap configuration during testing. Similarity scores include cosine and soft cosine similarity scores.

Table 1 List of all behavior options (full names and abbreviations) for MetaMap[22,23]

Abbreviations	Full names	Description
y	Use Word Sense Disambiguation	Eliminate word ambiguity, slower
700	Threshold (-r) set as 700	Only the outputs equal to or above the threshold will be displayed.
conj	Turn on Conjunction Processing	Recombine phrases separated by a conjunction, slower recall increases
10 (prune)	Prune Threshold set as 10	Specify the maximum number of candidates for mapping
t	No Text Tagging	Do not use part-of-speech tagging in parsing
u	Unique Acronym/Abbreviation Variants Only	Limit acronym/abbreviation variants only to the unique expansions
No prune	Disable Pruning	Disable pruning-candidate concepts
d	No Derivational Variants	Do not use any derived variations in the computation of word variants
i	Ignore Word Order	Process standard and nonstandard written English, recall increases
a	Allow Acronym/Abbreviation Variants	Use any acronym/abbreviation variants; they are less reliable.
all_derivational_variants	D	Allow using all derived variations, not only adjectives or nouns.
l	Allow Large N	Enable retrieval of two-character words >4,000 times or one-character words >2,000
m[a]	No Mappings	Disabled displays of mapping can only be used to show candidates.
negex[a]	Enable NegEx	Negated UMLS concepts will be displayed from the input.
Y[a]	Prefer Multiple Concepts	MetaMap scores higher with mapping more concepts versus fewer concepts
b[a]	Compute/Display All Mappings	MetaMap displays all mappings rather than only top-rated mappings.
Q[a]	Composite phrases	Enable MetaMap to composite longer phrases from smaller ones

Abbreviation: UMLS, Unified Medical Language System.
[a]Not used in our experiments.

Measurements

Cosine similarity measures the similarity between two nonzero vectors, representing terms or documents.[20] The cosine similarity value is based on the relative angle between the two vectors, with −1 representing 180° and 1 representing 0°. The equation used to calculate the cosine similarity score is described in ►Supplementary Appendix C.[20,21]

Soft cosine similarity is a probabilistic similarity measure that extends cosine similarity. In addition, soft cosine similarity considers the similarity of pairs of features between vectors. We used both cosine similarity and soft cosine similarity scores in the experiments because soft cosine similarity is more generalizable, particularly for NLP tasks. In the implementation, cosine similarity scores were calculated via a standard formula, and soft cosine similarity scores were computed using the gensim library 4.0.1 (softcosinesimilarity). The formula used to calculate soft cosine similarity is described in ►Supplementary Appendix C.[21,24]

During the evaluations, we considered and used exact matches and similar matches based on recommendations from Friedman and Hripcsak on evaluations in NLP projects.[25] ►Supplementary Appendix C presents the definitions of each match, percentages of exact matches, similar matches, and missing gold-standard terms, and their formulas. The percentages, based on the similarity scores of all identified entities among 40 abstracts, were used as indicators of MetaMap performance for every configuration.

Statistical Analysis

The current pipeline's similarity scores were manually spot-checked and measured with precision, recall, and an F-measure (when $\beta = 1$) (i.e., precision and recall were weighted as equally important). During manual annotations, the agreement between annotators was measured with Cohen's kappa rate.[26] Because there was only overlap between annotators 1 and 2 and annotators 1 and 3, but no overlap between annotators 2 and 3, we measured the agreement between the two raters using Cohen's kappa.

Paired two-sample *t*-tests were used to test the hypothesis that the percentages of exact matches calculated using cosine similarities and soft cosine similarities were identical for each MetaMap configuration. The same procedure was

Table 2 Crosswalk of combinations of two behavior options used in the MetaMap configuration experiments

Abbreviations	y	700	conj	10 (prune)	t	u	No prune	d	i	a	all_derivational_variants	l
y	–	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
700	✓	–	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
conj	✓	✓	–	✓	✓	✓	✓	✓	✓	✓	✓	✓
10 (prune)	✓	✓	✓	–	✓	✓	–	✓	✓	✓	✓	✓
t	✓	✓	✓	✓	–	✓	✓	✓	✓	✓	✓	✓
u	✓	✓	✓	✓	✓	–	✓	✓	✓	–	✓	✓
No prune	✓	✓	✓	–	✓	✓	–	✓	✓	✓	✓	✓
d	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	–	✓
i	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	✓
a	✓	✓	✓	✓	✓	–	✓	✓	✓	–	✓	✓
all_derivational_variants	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	–	✓
l	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–

Abbreviations: ✓, used as a configuration of MetaMap; –, not applicable.
Note: Refer to ►**Table 1** for additional abbreviations and full names.

used to compare the percentages of missing gold-standard terms calculated using cosine and soft cosine similarity scores for each MetaMap configuration. We selected $p < 0.001$ as the significance level.

We used this experiment design because the 40 annotated abstracts were randomly selected from the pool of 3,187 abstracts. We then used two methods (cosine similarity and soft cosine similarity) for calculations and compared the results via *t*-tests. We are doing this because methods 1 and 2 are independently implemented in all 40 abstracts, and there are no interactions whatsoever. Then, we used a *t*-test to detect the differences between the two methods per configuration.

Environmental Setting

We downloaded MetaMap (2020AA) and installed it locally. We used Palmetto Cluster, Clemson University's high-performance computing resource, to conduct the experiments. ►**Supplementary Appendix C** presents the default settings for MetaMap and the detailed parameters used for the word2vec and phrase models. The programs and model training were written in Python. PostgreSQL 13 was used as the database management system to host the raw data and candidate concepts identified with the automatic pipeline. All abstracts were extracted and downloaded using PubMed EFetch. ►**Supplementary Appendix D** lists the semantic networks selected for UMLS.

Manual Annotation

We randomly selected 44 abstracts for manual annotation and referenced the methodologies described in active learning.[27,28] First, three researchers independently annotated the abstracts and discussed the initial annotation to reach a consensus. At least two researchers evaluated each abstract. ►**Supplementary Appendix E** explains the principles used during the annotation and discussion. Examples of identified concepts via manual annotation included *operation, CDSS*, and *service model*.

Results

A total of 3,187 abstracts were used during training, and 40 abstracts were annotated and used for testing. The 40 annotated abstracts included 422 sentences (10.55 sentences/abstract), 9,375 words, and 1,052 gold-standard terms (2.49 entities/sentence). For the manual annotation, the Cohen's kappa rate between annotators 1 and 2 was 0.93, and between annotators 1 and 3, it was 0.73.

The percentages of exact matches, similar matches, and missing gold-standard terms based on the similarity scores were computed as indicators of MetaMap performance for each configuration. ►**Table 3** presents the percentages when using one option (i.e., 12 configurations), and ►**Table 4** shows the percentages when using two options (i.e., 63 configurations).

We found that the percentages of exact matches and missing gold-standard terms between cosine and soft cosine similarities for each MetaMap configuration differed significantly ($p < 0.001$) in paired two-sample *t*-tests (►**Tables 3** and **4**) except for one combination of two options: all_derivational_variant and when threshold set at 700 (ts_r_700, $p < 0.01$). The percentages calculated using soft cosine similarity scores exceeded those calculated using cosine similarity scores for each corresponding category.

We calculated the average precision, recall, and F-measure ($\beta = 1$) for exact matches, similar matches, and missing terms (►**Table 5**) via a manual spot-check. We noticed that (1) one behavior option generated the best result when using soft cosine similarity, and (2) the combination of two behavior options generated better results than one behavior option when using cosine similarity scores. ►**Fig. 3**, ►**Fig. 4**, and ►**Table 6** provide the sample results from our

Table 3 Percentages of exact matches, similar matches, and missing gold-standard terms calculated by cosine similarity or soft cosine similarity scores when using one option in MetaMap[a]

MetaMap option (one)	Percentages calculated based on cosine similarity scores			Percentages calculated based on soft cosine similarity scores		
	Exact match	Similar match	Missing term	Exact match[b]	Similar match	Missing term[b]
y	0.73	0.17	0.1	0.78	0.04	0.18
700	0.6	0.21	0.19	0.65	0.04	0.3
conj	0.72	0.17	0.11	0.78	0.04	0.18
10 (prune)	0.72	0.17	0.12	0.77	0.04	0.19
t	0.73	0.17	0.1	0.79	0.04	**0.17**
u	0.73	0.17	0.1	0.78	0.04	0.18
No prune	0.73	0.17	0.1	0.78	0.04	0.18
d	**0.74**	0.17	**0.09**	**0.79**	0.04	**0.17**
i	0.73	0.17	0.1	0.78	0.04	0.18
a	**0.74**	0.17	0.1	**0.79**	0.04	**0.17**
all_derivational_variants	0.73	0.17	0.1	**0.79**	0.04	0.18
l	0.73	0.16	0.11	**0.79**	0.04	0.18

[a]Bold text indicates the best results.
[b]$p < 0.001$ for differences between percentages calculated by cosine and soft cosine similarity scores per configuration by paired two-sample *t*-tests.
Note: no_prune is the default setting for MetaMap. Refer to ►**Table 1** for the abbreviations and full name for each option.

experiments. ►**Fig. 3** shows the comparison of results of eight abstracts (two sets of performance: highest and lowest percentages of exact matches) on correctly identified entities and gold standards during the manual spot-check. ►**Fig. 4** shows a comparison of correctly identified missing entities and gold standards. ►**Table 6** presents examples of identified terms, gold-standard terms, and the corresponding similarity scores.

Discussion

Implications and Interpretations of Our Results

Our study shares a *methodology* and codes that provide a framework and reference point for others who need MetaMap for their PARTICULAR tasks. Our systematic strategy to configure MetaMap can provide objective evidence that complements the current practices of using principles, intuitions, and experience for configurations. Optimizing MetaMap is one of the first and critical steps for many biomedical and health NLP tasks. Our manuscript demonstrates a data-driven approach to optimizing MetaMap based on performance measures.

Each percentage for exact matches, similar matches, and missing gold-standard terms can be especially valuable in providing an overall performance indicator in comparing MetaMap performance. These percentages can be used as an overall measure for other NLP or information retrieval tasks if such tasks are based on similarity scores. Although each of these parameters critically contributes to the percentage calculation, users can feel overwhelmed when considering all the similarity scores without percentages. We believe that our systematic evidence and data-driven approach help guide MetaMap configurations, creating a more *accurate and efficient process* to complete NLP tasks or data mining projects in the biomedicine and health fields.

Notably, one optimal configuration for MetaMap probably cannot be used for *any* focused area. Thus, for each subdomain to which MetaMap will be applied, the configuration will require *recalibration* based on systematic comparisons. While optimizing MetaMap configurations, we suggest that users make decisions based on the computation results and their preferences related to the tasks. We used both cosine similarity scores and soft cosine similarity scores to generate the overall percentages in this study. Based on the statistical tests, the percentages calculated using soft cosine similarity scores significantly exceeded those for cosine similarity scores for the exact matches and missing gold-standard terms for almost every configuration, except for one. Although these results suggest that the soft cosine similarity scores are better, the rates of missing gold-standard terms also require consideration. We noticed that such rates were almost twice as high when using soft cosine similarity scores across configurations compared with cosine similarity scores. We used cosine similarity scores for follow-up tasks, because we needed a low rate for missing gold standards (i.e., true negative rate) for our larger study (i.e., identifying entities from the literature for CDSS ontology).

In a previous pilot study, we assessed MetaMap performance in processing clinically actionable genomics texts.[29] This study expanded the pilot work to a larger scale, in a different area of focus, and with a more systematic approach. Specifically, we systematically compared exhaustive combinations and permutations of the two behavior options for MetaMap. Interestingly, not all results of this study corroborated the results of our pilot study. For example, in the pilot study, we observed that combining different

Table 4 Percentages of exact matches, similar matches, and missing gold-standard terms calculated by cosine similarity or soft cosine similarity scores when using two options in MetaMap[a]

MetaMap option (two)	Percentages calculated based on cosine similarity scores			Percentages calculated based on soft cosine similarity scores		
	Exact match	Similar match	Missing term	Exact match[b]	Similar match	Missing term[b]
l_y	0.73	0.17	0.11	0.79	0.04	0.18
conj_no_tagging	0.73	0.17	0.1	0.79	0.04	**0.17**
r_700_i	0.6	0.21	0.19	0.65	0.04	0.3
conj_i	0.72	0.17	0.11	0.78	0.04	0.18
all_derivational_variants_r_700	0.61	0.2	0.18	0.66[c]	0.05	0.3
no_tagging_r_700	0.56	0.22	0.22	0.62	0.04	0.34
conj_no_prune	0.72	0.17	0.11	0.78	0.04	0.18
prune_10_r_700	0.59	0.21	0.2	0.65	0.04	0.31
prune_10_no_tagging	0.72	0.17	0.11	0.78	0.04	0.18
prune_10_all_derivational_variants	0.73	0.17	0.11	0.77	0.04	0.19
no_tagging_u	0.73	0.17	0.1	0.79	0.04	**0.17**
no_derivational_variants_r_700	0.6	0.2	0.19	0.66	0.04	0.3
prune_10_a	0.72	0.17	0.11	0.78	0.04	0.19
prune_10_u	0.72	0.17	0.12	0.77	0.04	0.19
no_derivational_variants_i	0.73	0.17	0.1	0.79	0.04	0.18
no_tagging_all_derivational_variants	**0.74**	0.17	**0.09**	0.79	0.04	**0.17**
no_tagging_a	**0.74**	0.17	**0.09**	0.79	0.04	**0.17**
conj_y	0.72	0.17	0.11	0.78	0.04	0.18
no_prune_u	0.73	0.17	0.1	0.78	0.04	0.18
all_derivational_variants_u	**0.74**	0.17	**0.09**	0.79	0.04	**0.17**
conj_prune_10	0.7	0.18	0.12	0.76	0.04	0.19
r_700_y	0.6	0.21	0.2	0.65	0.04	0.3
conj_all_derivational_variants	0.73	0.17	0.1	0.78	0.05	**0.17**
prune_10_l	0.71	0.18	0.12	0.76	0.04	0.2
conj_r_700	0.57	0.22	0.21	0.63	0.04	0.33
no_tagging_no_derivational_variants	0.73	0.17	0.1	0.79	0.04	**0.17**
no_derivational_variants_y	0.73	0.17	0.1	0.79	0.04	0.18
a_y	0.73	0.17	0.1	0.79	0.04	**0.17**
no_tagging_l	**0.74**	0.17	0.1	0.79	0.04	**0.17**
u_l	0.73	0.16	0.11	0.79	0.04	0.18
conj_no_derivational_variants	0.72	0.17	0.11	0.78	0.04	0.18
all_derivational_variants_l	**0.74**	0.16	0.1	0.79	0.04	**0.17**
prune_10_i	0.72	0.17	0.12	0.77	0.04	0.19
l_i	0.73	0.16	0.11	0.79	0.04	0.18
no_tagging_y	0.73	0.17	0.1	0.79	0.04	**0.17**
a_r_700	0.61	0.21	0.19	0.66	0.04	0.3
no_prune_i	0.73	0.17	0.1	0.78	0.04	0.18
all_derivational_variants_i	**0.74**	0.17	**0.09**	0.79	0.04	**0.17**
no_prune_r_700	0.6	0.21	0.19	0.65	0.04	0.3
u_y	0.73	0.17	0.1	0.78	0.04	0.18

Table 4 (*Continued*)

MetaMap option (two)	Percentages calculated based on cosine similarity scores			Percentages calculated based on soft cosine similarity scores		
	Exact match	Similar match	Missing term	Exact match[b]	Similar match	Missing term[b]
prune_10_y	0.71	0.17	0.12	0.77	0.04	0.19
all_derivational_variants_a	**0.74**	0.16	**0.09**	0.79	0.04	**0.17**
no_prune_l	0.73	0.16	0.11	0.79	0.04	0.18
no_tagging_i	0.73	0.17	0.1	0.79	0.04	**0.17**
prune_10_no_derivational_variants	0.72	0.17	0.11	0.77	0.04	0.19
conj_l	0.73	0.17	0.11	0.78	0.04	0.18
no_prune_a	**0.74**	0.17	0.1	0.79	0.04	**0.17**
u_r_700	0.6	0.21	0.19	0.65	0.04	0.3
a_i	**0.74**	0.17	0.1	0.79	0.04	**0.17**
no_prune_all_derivational_variants	**0.74**	0.17	**0.09**	0.79	0.04	**0.17**
conj_u	0.72	0.17	0.11	0.78	0.04	0.18
no_prune_no_derivational_variants	0.73	0.17	0.1	0.79	0.04	0.18
i_y	0.73	0.17	0.1	0.78	0.04	0.18
l_r_700	0.6	0.2	0.19	0.65	0.04	0.3
u_i	0.73	0.17	0.1	0.78	0.04	0.18
no_prune_no_tagging	0.73	0.17	0.1	0.79	0.04	**0.17**
all_derivational_variants_y	**0.74**	0.17	0.1	0.79	0.04	**0.17**
no_prune_y	0.73	0.17	0.1	0.78	0.04	0.18
conj_a	0.73	0.17	0.1	0.79	0.05	**0.17**
no_derivational_variants_l	0.73	0.16	0.11	0.79	0.04	**0.17**
no_derivational_variants_a	**0.74**	0.16	0.1	**0.8**	0.04	**0.17**
no_derivational_variants_u	0.73	0.17	0.1	0.79	0.04	0.18
a_l	**0.74**	0.16	0.1	0.79	0.04	**0.17**

[a]Bold text indicates the best results.
[b]$p < 0.001$, [c]$p < 0.01$ for differences between percentages calculated by cosine and soft cosine similarity scores by paired two-sample *t*-tests.
Note: Refer to ►**Table 1** for each option's abbreviations and full name.

Table 5 Manual spot-check results for similarity scores with different MetaMap configurations

MetaMap configuration	Types of match	Precision	Recall	F-measure
Cosine-one option	Exact and similar matches	0.512	0.553	0.532
	Missing terms	1	0.306	0.469
Soft cosine-one option	Exact and similar matches	0.69	0.958	0.802
	Missing terms	1	0.842	0.914
Cosine-two options	Exact and similar matches	0.571	0.903	0.699
	Missing terms	1	0.64	0.78
Soft cosine-two options	Exact and similar matches	0.56	0.875	0.683
	Missing terms	1	0.647	0.786
Average	*Exact and similar matches*	*0.587*	*0.82*	*0.684*
	Missing terms	*1*	*0.527*	*0.69*

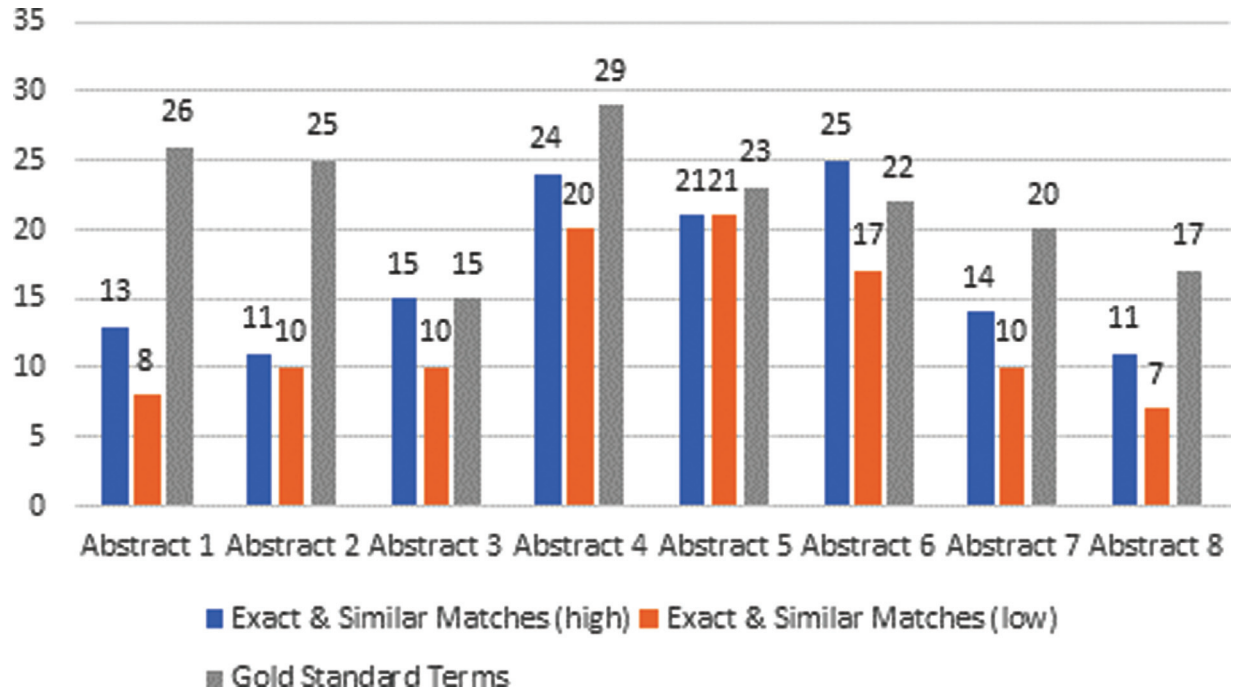

Fig. 3 Comparison of correctly identified entities and gold standards among eight abstracts with two sets of performance (high and low) per abstract during the manual spot-check.

behavior options, especially the best-performing options, outperformed singular options. However, in this study, when using cosine similarity scores, the precision, recall, and F-measure of the two behavior options outperformed one behavior option. However, the default behavior option, one behavior option, and two behavior options did not appear to generate significantly different percentages. Therefore, we did not assess the additional combination of options.

We used precision, recall, and F-measure to validate the similarity scores in the experiments. Although these measures could be higher, we believe the results were reasonable. For example, the average precision was 0.587, and the average F-measure was 0.684 for exact and similar matches.

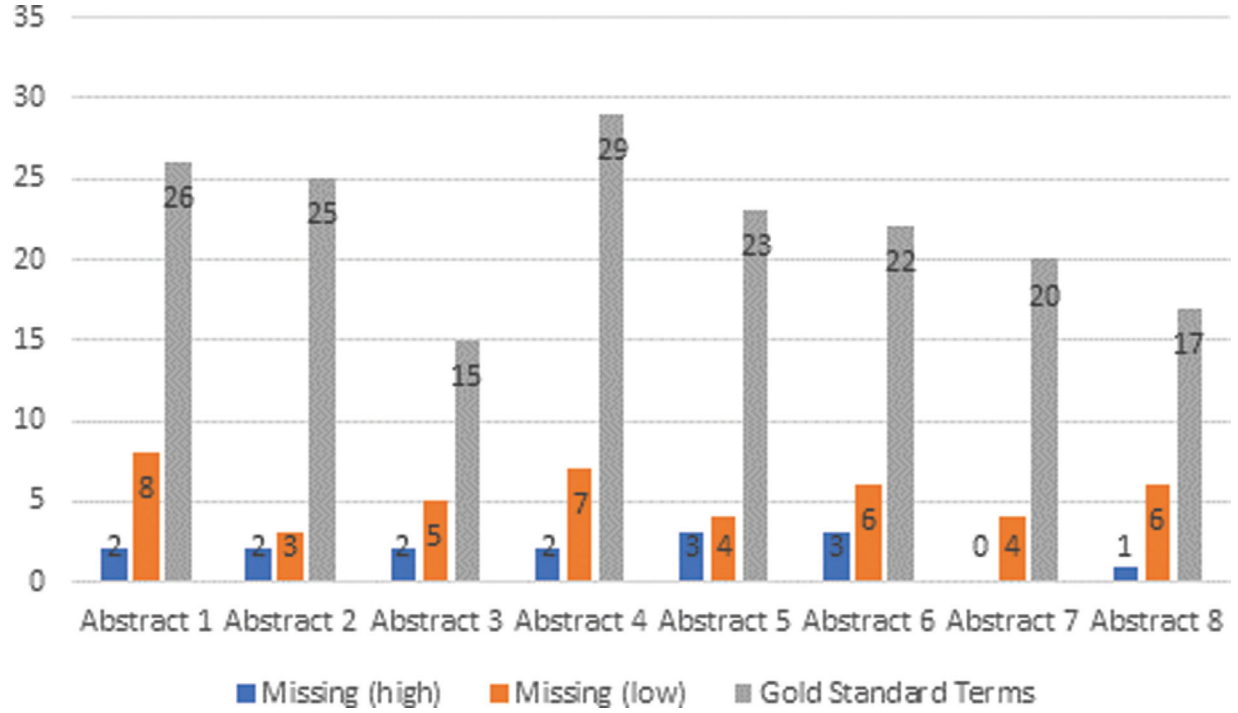

Fig. 4 Comparison of correctly identified missing entities and gold standards among eight abstracts with two sets of performance (high and low) per abstract.

Table 6 Examples of identified entities and the corresponding similarity scores

	Identified entity by pipeline	Gold-standard term	Cosine similarity	Soft cosine similarity
Exact match	Intelligent information system	Intelligent health information system	N/A	1
	Ontology	Ontology engine	0.9	N/A
Similar match	Record	Electronic health record	N/A	0.79
	Domain	Domain expert	0.81	N/A
Missing	N/A	Scalable	N/A	N/A
	N/A	Integration	N/A	N/A

Abbreviation: NA, not applicable.

Because our tasks are much more challenging than binary judgments, we still did not achieve 100% agreement, even after discussion among human annotators. That is, the nature of this work differs from the annotation of a binary medical diagnosis from a patient record. In our study, various implementation cases could exist for the CDSS ontology, which ultimately dictates what entities can be included in the ontology.

The manual spot-check results show encouraging evidence, but there is room for improvement. Meanwhile, we observed that these results were based on match results (i.e., identifying the entity and matching it to the gold-standard terms). If we consider only the results identified by the models, the performance would be slightly higher. We compared our results with other studies that evaluated name entity recognition[10,30–32]; although our results were lower, they were comparable, even though we focused on a different area and used different systems and methodologies.

The generalizability of this work can be demonstrated on several levels: principles, exact methods, and results. Others can completely repeat the work described in this manuscript at the principal and methods levels. The methods, criteria, steps, and flow can be reused easily. The codes we shared via GitHub can easily be adapted to a different topic. Our results can also be used as a reference point too. Although our results cannot be used as a universal benchmark for any topic area, we feel that the methods, principles, and codes provide good levels of generalizability. In summary, the key contribution of the work is to demonstrate a systematic way to test NLP tasks' performances using various combinations of MetaMap options. The results can guide the configuration and optimization of MetaMap, that is, to provide a case study to demonstrate the use of systematic testing results to guide tool configuration and optimization in completing informatics tasks. The method we demonstrated complements the current practice of relying on experience, principles, and intuition. Although MetaMap has been broadly used in NLP tasks in biomedicine, we did not find any published papers conducting similar work. The reality indicates that such systematic comparison is not standard practice yet.

Discussion of the Model

In this study, our goal was to determine the optimal configuration of MetaMap by comparing the corresponding performance of similarity scores with different configurations. We believe that MetaMap and the word2vec and phrase models are reasonable choices for our tasks. During training, the phrase model creates phrases (bigrams and trigrams) from text with words that occur together. Once all the text is processed by the phrase model, the phrases are passed to the word2vec model to create vectors. Then, cosine and soft cosine similarity scores were used to calculate the distances between the results generated from the pipeline and those from the gold-standard terms. We did not compare the use and nonuse of the phrase model. Our intuition was to build the phrases as entities that are solely multi-word terms, which is closer to reality. Building phrases essentially synergizes the words that frequently occur as normalized pointwise mutual information (npmi)[33] in phrases, and not all words were included. This process (1) reduces the vocabulary by building phrases that merge a few words that frequently occur together and (2) builds bigrams or trigrams with the npmi algorithm, which can keep words together if they frequently occur together.

Our Study versus Traditional Machine Learning Studies

Our methodology differs from traditional machine learning methodologies. For example, our methodology includes training, testing steps, and a manual annotation set, whereas classic machine learning usually includes training datasets, testing datasets, and cross-validation. Our methodology uses the unsupervised training step for the phrase and word2vec models to learn vocabularies and contexts from the whole set of abstracts. We use the testing step to (1) evaluate MetaMap performance based on different configurations with multiple measurements and (2) uncover the unsupervised training results using a small percentage of abstracts with gold standards.

From a classic machine learning perspective, we might overfit word2vec and phrase models. However, this study aimed not to show the models' accuracy, as occurs in many traditional machine learning studies. However, we used the models to generate quantitative values to compute and compare the MetaMap performance with different configurations. If the models are a bit "overfit," they are "overfit" for both cosine and soft cosine similarity scores; the percentages calculated from the cosine and soft cosine similarity scores

will still be comparable. In future work, we plan to use "newly added abstracts," that is, newly published abstracts or papers from PubMed and nonduplicate abstracts from the Association for Computing Machinery Digital Library to evaluate the pipeline's performance.

Limitations and Future Directions

One limitation of this study is the source vocabularies included in UMLS. UMLS does not include vocabularies yet for health information technology (IT), which negatively affects MetaMap performance for our particular task. However, we must recognize that there is no existing vocabulary in the health IT field yet; therefore, this limitation is not due to UMLS. We hope that our work with the CDSS ontology will create a starting point to curate a vocabulary in health IT.

This study used 40 annotated abstracts (422 sentences) to calculate the similarity scores. Wei et al[28] suggested that such a corpus size is at the lower end of sample sizes of annotated sentences based on their extensive experiments. Although we recognize that a larger scale of annotation and manual spot-check may provide additional insights, our sample size seems to be within a reasonable scope. In the future, we can increase the sample size. We could also develop a new pipeline that processes full-text papers or assesses different focused areas (e.g., e-prescribing) to determine whether it can generate similar results to this study.

This study assessed a limited number of options and is a starting point for systematically comparing MetaMap performance. Future work might compare MetaMap performance using additional combinations and transformations of options based on intense error analysis under different settings.[34] In this study, for tuning the word2vec model, we assessed different window sizes. We checked the top 10 cosine matching terms (using word2vec.most_similar function of gensim). We assessed window_size as 2, 5, 7, and determined it to be 5. Similarly, for the phrase model, we assessed the minimum count as [1,**3**,5] and threshold as [0.4, **0.5**, 0.6] (the bold numbers were used as the final hyperparameter values due to better results while training the model). This tuning process can be performed more deeply with intensive error analysis or an additional candidate for parameter values and combinations.

Conclusions

In summary, we demonstrated a systematic strategy for configuring and optimizing MetaMap performance, a valuable tool for NLP in the biomedical and health fields. Our method provides objective and accurate evidence that complements current practices of using principles, experience, and intuition to guide MetaMap configurations. Combining objective evidence, human experience, and expertise supports a better methodology than using any single strategy for completing complicated tasks. Although our work focuses on the CDSS literature, the methodology, workflow, measurements, and codes can be applied and referenced for other focused areas.

Funding

This work is supported by the National Institute of General Medical Sciences of the National Institutes of Health under Award No. R01GM138589 and partially under P20 GM121342. We acknowledge Clemson University for the generous allotment of computing time on the Palmetto Cluster.

Conflict of Interest

None declared.

Ethical Approval

Our experiments only used publicly accessible literature without patient data. Therefore, approval by an institutional review board was not required.

References

1 Chen Y, Elenee Argentinis JD, Weber G. IBM Watson: how cognitive computing can be applied to big data challenges in life sciences research. Clin Ther 2016;38(04):688–701
2 Ferrucci D, Levas A, Bagchi S, et al. Watson: beyond jeopardy!. Artif Intell 2013;199–200:93–105
3 Chen W, Hu Y, Zhang X, et al. Causal risk factor discovery for severe acute kidney injury using electronic health records. BMC Med Inform Decis Mak 2018;18(Suppl 1):13
4 Zhou L, Blackley SV, Kowalski L, et al. Analysis of errors in dictated clinical documents assisted by speech recognition software and professional transcriptionists. JAMA Netw Open 2018;1(03): e180530
5 Wang J, Lavender M, Hoque E, Brophy P, Kautz H. A patient-centered digital scribe for automatic medical documentation. JAMIA Open 2021;4(01):b003
6 Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc AMIA Symp 2001: 17–21
7 Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc 2010; 17(03):229–236
8 Savery ME, Rogers WJ, Pillai M, Mork JG, Demner-Fushman D. Chemical entity recognition for MEDLINE indexing. AMIA Jt Summits Transl Sci Proc 2020;2020:561–568
9 Chiaramello E, Paglialonga A, Pinciroli F, Tognola G. Attempting to use MetaMap in clinical practice: a feasibility study on the identification of medical concepts from italian clinical notes. Stud Health Technol Inform 2016;228:28–32
10 Chapman WW, Fiszman M, Dowling JN, Chapman BE, Rindflesch TC. Identifying respiratory findings in emergency department reports for biosurveillance using MetaMap. Stud Health Technol Inform 2004;107(Pt 1):487–491
11 Peng J, Zhao M, Havrilla J, et al. Natural language processing (NLP) tools in extracting biomedical concepts from research articles: a case study on autism spectrum disorder. BMC Med Inform Decis Mak 2020;20(Suppl 11):322
12 Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res 2004;32 (Database issue):D267–D270
13 Pires DF, Teixeira CAC, Ruiz EES. A UMLS interoperable solution to support collaborative diagnosis decision making over the internet. Paper presented at: Proceedings of the 2008 ACM symposium on Applied computing; 2008; Fortaleza, Ceara, Brazil. Accessed June 10, 2022 at: https://doi-org.libproxy.clemson.edu/10.1145/1363686.1364009
14 Warren JJ, Matney SA, Foster ED, Auld VA, Roy SL. Toward Interoperability: a new resource to support nursing terminology standards. Comput Inform Nurs 2015;33(12):515–519

15 Bhupatiraju RT, Fung KW, Bodenreider O. MetaMap Lite in Excel: biomedical named-entity recognition for non-technical users. Stud Health Technol Inform 2017;245:1252
16 Demner-Fushman D, Rogers WJ, Aronson AR. MetaMap Lite: an evaluation of a new Java implementation of MetaMap. J Am Med Inform Assoc 2017;24(04):841–844
17 Pratt W, Yetisgen-Yildiz M. A study of biomedical concept identification: MetaMap vs. people. AMIA Annu Symp Proc 2003; 2003:529–533
18 Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. Proceedings of Workshop at ICLR, 2013. https://arxiv.org/pdf/1301.3781.pdf
19 Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13). Curran Associates Inc., Red Hook, NY, USA, 3111–3119
20 Manning CD, Raghavan P, Schütze H. An Introduction to Information Retrieval. Cambridge: Cambridge University Press; 2009
21 Sidorov G, Gelbukh A, Gómez-Adorno H, Pinto D. Soft similarity and soft cosine measure: similarity of features in vector space model. Comput Sist 2014;18(03):491–504
22 Lang Fc-M. MetaMap Usage Notes. 2016. Accessed Aug 30, 2021 at: https://metamap.nlm.nih.gov/Docs/MM_2016_Usage.pdf
23 National Library of Medicine. MetaMap-A tool for recognizing UMLS concepts in text. Accessed Sept 27, 2019 at: https://metamap.nlm.nih.gov
24 Novotný V Implementation notes for the soft cosine measure. Paper presented at: The 27th ACM International Conference on Information and Knowledge Management; 2018; Torun, Italy. Accessed June 10, 2022 at: https://doi.org/10.1145/3269206.3269317
25 Friedman C, Hripcsak G. Evaluating natural language processors in the clinical domain. Methods Inf Med 1998;37(4-5):334–344
26 McHugh ML. Interrater reliability: the kappa statistic. Biochem Med (Zagreb) 2012;22(03):276–282
27 Chen Y, Lask TA, Mei Q, et al. An active learning-enabled annotation system for clinical named entity recognition. BMC Med Inform Decis Mak 2017;17(Suppl 2):82
28 Wei Q, Chen Y, Salimi M, et al. Cost-aware active learning for named entity recognition in clinical text. J Am Med Inform Assoc 2019;26(11):1314–1322
29 Merchant O, Tellur S, Jing X. A pilot evaluation of the performance of metamap for processing clinical actionable genomics texts. AMIA Summit 2021, Virtual, 2021. P857
30 Marrero M, Sánchez-Cuadrado S, Lara JM, Andreadakis G. Evaluation of named entity extraction systems. Research in Computing Science 2009;41:47–58
31 Tsai RT-H, Wu S-H, Chou W-C, et al. Various criteria in the evaluation of biomedical named entity recognition. BMC Bioinformatics 2006;7(01):92
32 Song H-J, Jo B-C, Park C-Y, Kim J-D, Kim Y-S. Comparison of named entity recognition methodologies in biomedical documents. Biomed Eng Online 2018;17(02, Suppl 2):158
33 Bouma G. Normalized (pointwise) mutual information in collocation extraction. 2009. Accessed June 10, 2022 at: https://svn.spraakdata.gu.se/repos/gerlof/pub/www/Docs/npmi-pfd.pdf
34 Divita G, Tse T, Roth L. Failure analysis of MetaMap Transfer (MMTx). Stud Health Technol Inform 2004;107(Pt 2):763–767