NLP project proposal
SLR - NieLeniweProjekty

First Author: Michał Gozdera
Warsaw University of Technology
01142172@pw.edu.pl

Second Author: Małgorzata Hadasz
Warsaw University of Technology
01156169@pw.edu.pl

Third Author: Krystian Kurek
Warsaw University of Technology
01121582@pw.edu.pl

supervisor: Anna Wróblewska
Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

Abstract

Nowadays, the rapid increase in knowledge and the amount of performed research in numerous domains (like Computer Science and Medicine) causes the need for solutions designed to automatically segregate, organize and find created papers and publications. Key aspect of such frameworks is to detect the topic of a given article, and connect the discovered subject with domain-specific concepts. Hence, the aim of our project is to address the question of finding semantic keywords for systematic literature overview. Namely, we propose different solutions to extract keywords from medical papers abstracts, tag these keywords with ontologies concepts and choose the best tags based on disambiguation techniques.

For keywords extraction, we focus mainly on BERTopic model, but other algorithms (like LDA) can possibly be compared.

One approach to keywords tagging is going to be performed with the use of NBCO annotator (standard and simple solution) while the other will be implemented from scratch with the use of words embedding concept (*word2vec*).

Tag disambiguation techniques will rely on Closest Sense method, adjusted to our problem statement.

To the best of our knowledge, there is currently no state-of-the-art solution combining three functionalities mentioned above, and taking advantage of the latest NLP solutions.

1 Introduction

Medicine scientific area is characterized by a rapid pace of creating new literature. Every year, numerous new papers in different medical domains are produced. On the other hand, reliable research requires authors to be familiar with current achievements, which in turns causes the need for effective and exact literature searching.

The scientific goal of our research is to create a solution that would help researchers and people interested in medical papers to find the most suitable scientific publications according to their needs.

The research question we want to address is whether the solution described above can be prepared with the use of the latest achievements in NLP domain. Our proposition is based on several techniques, including keywords extractors, ontology-based taggers and disambiguation algorithms.

2 Current solutions and state-of-the-art

Currently available solutions are not specifically directed into the aim we presented. There exist state-of-the-art solutions performing specific parts of what we are going to implement, but the entire process itself is not well investigated in our opinion.

Regarding keywords extraction, Latent Dirichlet Allocation - LDA (Blei et al., 2003) is one of the state-of-the-art algorithms, however currently it is usually replaced by models utilizing modern words embedding techniques, like BERTopic (Grootendorst, 2022b).

For a long time, for ontology based tagging in medical data, simple solutions, like NCBO annotator (Jonquet et al., 2009) were used. Recently, more sophisticated approaches (like ScispaCy (Neumann et al., 2019)) appeared. What is more, words embedding idea is getting more and more interest in the NLP filed, actually be-

ing the current state-of-the-art word representation (mainly because of its high performance and important properties, like preserving semantic meaning). However, to the best of our knowledge, there is no any state-of-the-art embedding based annotator provided for the use with specific medical ontologies.

Ontology tag disambiguation is according to our research the least explored part of the solution. There are not many papers approaching this topic, most of them treating disambiguation as a side part of other solutions ((Leaman and Lu, 2016), (Bindelli et al., 2008a)). An algorithm that seems to fit the needs of our solution best is Closest Sense method (Alexopoulou et al., 2009).

3 Significance of the project

According to current state-of-the-art research it is clear, that there is no obvious solution for semantic keywords for systematic literature reviews problem available. Based on our research, there exist models and algorithms than can be successfully modified and then combined together to create a well-performing method tackling the scientific problem we describe.

The pioneering nature of our project include not only combing existing solutions into one, designed for specific problem, but also innovative modifications of existing methods and implementing our own ideas from scratch.

We aim to contribute to the research filed in two areas:

- research resulting in creating a method of extracting semantic keywords for systematic literature reviews, based on recent NLP achievements,
- developing an actual solution and implementation that can be successfully incorporated into research community and improve the quality and speed of research work.

4 Concept and work plan

In this section we describe the project analysis and time scheduling. The main milestones and goals are shown. Moreover, we present the results of the preliminary research and risk analysis.

4.1 Project activities and timeline

We divided our project into 3 main parts, that are presented in the table 1.

Date	Stage name	Description
4.11.2022	Project proposal	literature review, solution concept and proposal
18.11.2022	Proof of concept	exploratory data analysis and preliminary machine learning models
9.12.2022	Final project	full solution and prepared product

Table 1: Project activity and timeline

4.2 Specific research goals

We establish the following research goals for the project:

- acquiring wide knowledge about current state-of-the art methods in NLP domain, especially in semantic keywords for systemic literature reviews field,
- testing different solutions currently available and adapting them to the above need,
- combining existing methods for keywords retrieval, keywords ontology annotating and tags disambiguation into one, working solution,
- creating new algorithms for keywords ontology annotating and tags disambiguation.

4.3 Results of preliminary research

Introductory research resulted in gathering knowledge about state-of-the-art methods than can be incorporated into our solution. They are described in section 2. Apart from reading about particular solutions, we also tested and verified existing implementations:

- Keywords extraction - the LDA algorithm is available in *sklearn* library (Buitinck et al., 2013); BERTopic can be found in *bertopic* package (Grootendorst, 2022a),
- Keywords tagging – for NCBO annotator, the REST API is available (Jonquet et al., 2021), so our solution is going to use it via HTTP connection; there is no concrete implementation of embedding-based annotations with medical ontologies that would satisfy our need, so this part will be implemented from scratch based on *word2vec* implementation (*gensim* package (Rehurek, 2022)),

- Tags disambiguation – since we are going to use a modified version of Closest Sense method, we are going to implement it from scratch.

The main result of the preliminary research is that currently available solutions (with quite a few modifications) should allow us to develop a fully usable method for extracting semantic keywords. However, there is not any specific algorithms pipeline (combining all above methods) available now. Creating one will be the goal of our project.

To develop and test our solution, we planned to use the MedMentions data set (Mohan and Li, 2019). This resource provides access to over 4 000 articles (titles and abstracts) published in PubMed. Each article is annotated with UMLS (Bodenreider, 2004) concepts by professional annotators with rich experience in biomedical content. We will treat those articles and annotations as the Gold Standard. In the MedMentions' release article, they have mentioned other relevant corpora that we can use in case of any problems with MedMentions.

In part of our solution we need to provide the ontology. We plan to test a few of them. The main is the one, that is used in the dataset, but we also decide to try different ones:

- *UMLS* (Bodenreider, 2004)
- *The Human Disease Ontology* (Sargsyan et al., 2020)
- *International Classification of Diseases, Version 10* (Möller et al., 2010).

Although, we might change them, if we find more suitable ones, or the aforementioned ontologies won't be satisfying.

4.4 Risk analysis

The risk analysis can be divided into 4 parts. The risk of the data, the risk of the algorithms, the time shortage, and the risk of the team.

The first one may include data leakage, data removal, or the change of data privacy policy. We don't use any vulnerable information (etc. personal numbers), therefore the risk of leakage is not high and if it occurs it wouldn't affect anyone. We make use of open-source data sets and ontologies. Consequently, we would be affected by their removal or the change of privacy policy. To prevent it, where possible, we use multiple ontologies and datasets. As a result of that, our algorithm can work with only part of those data.

The second hazard is the risk of the algorithms. As in the previous point, the framework owners might change the rights to the algorithm (make it private), or the tested approach might not bring the expected results. The former might be solved by using other similar frameworks, or by developing only the part of the project. Another approach might be to develop a similar algorithm from scratch. However, it would significantly enlarge our project and might lead to a lack of time for the rest of the development. The latter might be caused by various factors. The computational power, that we have, might be too low and would not allow us to train the algorithm for the desired amount of time. The pre-trained models might not be suited for the kind of data we are using, the frameworks might not work with themselves properly or the results might not be satisfactory due to other, unrecognized causes.

The next risk is time shortage. Due to various factors, we might not have enough time to finish the project. The factors might be those mentioned in this analysis, underestimating the scope of the project, or other previously unrecognized ones. In case of a time shortage, we might develop only part of the solution and finish the project in the next assignment.

The last risk is linked to the team. We might be affected by a mistake made by a team member, experience communication issues, or a part of the team might want to leave the project. The first one is for instance deletion of an important part of the project, to prevent it we use a shared repository and commit changes after every important change. Moreover, we control the work of our coworkers to detect possible mistakes.

The aforementioned issues are the main ones, that we might experience. Other, unrecognized issues might occur. We would undertake all possible actions to prevent them.

5 Approach & research methodology

Our approach and research methodology consist of several steps.

Firstly, we aim to perform a thorough research in both general NLP and methods specific to our problem. This part of the project is almost finished

and this report states its results. Of course, during the next phases of the project, subsequent research activities probably occur, since the development process is an iterative task.

The second part of the problem investigation is to test various implementations of currently available methods. As described above, this part is also done.

Next, we aim to prepare a Proof of Concept (PoC) solution that will utilize some of concepts included in previous sections. It will be probably composed of ready-to-use solutions, like BERTopic + NCBO annotator. Its aim is to illustrate the way system will work.

Then, we intend to prepare the final solution, including all methods described in previous sections. Since the project incorporates a research rather than development approach, we reserve that some of the planned methods may change. Each stage of the project is going to be presented in front of other researchers working on similar projects as well as the project supervisor. Consultations with the supervisor are planned through all stages.

6 Methods, techniques, devices to be used in research

In this section we describe methods and techniques used in developing the project solution.

6.1 Keyword extraction

To detect the main concepts of the given documents, we decided to use Topic Modeling. It is an unsupervised machine learning method, that scans a set of documents and clusters them into groups represented by similar abstract topics. The conventional technique LDA (Blei et al., 2003) treats a document as a bag-of-words. Consequently, it loses the context and the order of the words. To prevent order loss and profit from the context of the given word, text embedding techniques have been used in various tasks. In recent years they became popular in the topic modeling field. Therefore, we decided to use BERTopic (Grootendorst, 2022b).

6.1.1 LDA

The Latent Dirichlet Allocation is a generative probabilistic model for finding hidden topics in the given corpora, proposed in (Blei et al., 2003). It makes a few assumptions:

1. topics are the statistically significant words in given corpora,

2. documents are a mixture of topics,

3. topics are a mixture of words.

Based on them, LDA calculates the probability density of topics in the document.

Before performing the algorithm pre-processing is needed, words need to be tokenized and a number of expected topics need to be given (in this work it is going to be denoted as Q). After that, the word-document matrix is created. This matrix is then divided into two matrices: document-topic and topic-word one.

LDA is an iterative process. In the first iteration, the randomly selected topics are assigned to each word. After that, LDA tries to optimize the results. In order to do this, it examines each word separately. Assuming that all assigned topics, apart from the current one, are correct. LDA tries to find the best topic for a given word. To do this it calculates 2 probabilities;

1. p1: proportion of words in a given document with a given topic (q),

2. p2: proportion of the documents in which the word (w) has the topic q assigned.

Using those probabilities, it detects the most relevant topic for a given word and reassigns it.

For each word in each document, the procedure is repeated, until a steady solution is found. At the end, the list of Q tuples containing the topic number and the list of most informative terms with their probability is given. LDA doesn't interpret topics, this step needs to be performed manually (it only provides the topic number and the informative words, user needs to add the topic description/ name if needed).

6.1.2 BERTopic

In this project we use the (Grootendorst, 2022a) BERTopic framework implementation.

To generate topic representation, BERTopic goes through 3 main steps.

First, it embeds documents in order to create their representation in vector space and compare their semantic meaning. As a default, it uses Sentence-BERT (SBERT) framework (Reimers and Gurevych, 2019a), which enables converting

sentences into vector representation using a pre-trained language model. SBERT is an extension to the traditional BERT model, for which calculating the sentence probability is a very time-consuming task. As authors of the (Reimers and Gurevych, 2019b) claim, by adding the pooling operation at the output of the BERT, the time of finding the most similar sentence pair in a collection of 10000 sentences was reduced from 65 hours to 5 seconds. Therefore, the BERTopic framework, by default, makes use of the SBERT model. It also allows using other pre-trained sentence embedding or custom models.

Subsequently, it performs clustering. Due to high space dimensionality, calculating the distance might become ill-defined. Therefore, to reduce dimensionality UMAP (McInnes et al., 2018) algorithm is used. The reduced embeddings are clustered using HDBSCAN (McInnes et al., 2017). The BERTopic framework allows changing both dimensionality reduction and clustering algorithms. The aforementioned techniques are used as default methods.

The last step is finding the topic representation. As a default, the modified TF-IDF procedure is used. The original procedure combines term and inverse document frequency:

$$W_{t,d} = tf_{f,d}\dot{l}og(\frac{N}{df_t}), \tag{1}$$

where $tf_{t,d}$ is the frequency of the term t in document d, N is the number of documents and df_t is a document frequency that shows how much information the term provided in the document. In BERTopic this procedure is generalized to clusters of documents. Firstly, all documents in the cluster are concatenated, then TF-IDF is modified and obtained by the formula:

$$W_{t,c} = tf_{f,c}\dot{l}og(1 + \frac{A}{tf_t}), \tag{2}$$

where tf_t, c is a frequency of the term t in the class c. C is concatenated into one document collection of the documents from the same cluster. Tf_t is a class frequency, measuring how much information the term provides to a class. By using the modified TF-IDF formula the importance of the words in a cluster, rather then in the document, is modeled.

6.2 Tagging tools

Keywords, extracted in the previous step, might incorporate different names to describe the same concepts. Therefore, to make use of them it is essential to perform mapping into existing ontologies. The ontology provides the standardized, homogenous, and informative concept, that describes the given keyword. The task of tagging the word with an entity existing in the ontology is called entity normalization, entity grounding, or entity categorization. As an example of the biomedical entity grounding, we will use the Onto-Biotope Ontology (Nédellec et al., 2018). Given the words "pediatric", "respiratory" and "children less than 2 years", we aim to find the appropriate tags in the ontology. For the first two words, the task is relatively simple. "Pediatric" ought to be linked to "pediatric patient" and "respiratory" to the "respiratory tract part". Both of those examples are lexically similar. In the last case, the linking is not that trivial. "Children less than 2 years" should be tagged as a "pediatric patient", even though the lexical similarity doesn't exist. The given example was derived from (Karadeniz and Özgür, 2019). Moreover, in the biomedical domain, the number of semantic categories is greater than the entities mentioned in available training data sets. For example, Onto-Biotope ontology consists of 2221 categories, while only 747 of them were mentioned in the training data set. Therefore, we decided to use unsupervised annotation techniques. In this section, we described two approaches tested in our solution. The first is NCBO tagger, which annotates data based mostly on direct string matching. The second, which uses word embeddings to link entities using word.

6.2.1 NCBO tagger

NCBO tagger (Jonquet et al., 2009) is a result of an initiative to construct a solution for annotating biomedical data with the use of a great number of ontologies. At the time of releasing the solution it used over 200 ontologies and this number is constantly increasing.

The way NCBO tagger works is simple, yet in many cases powerful enough. It uses a few steps to tag each token of an input free-text.

First of all, a direct string matching is performed. The dictionary of ontologies concepts is used for this purpose. It is constructed by pooling all concept names or other string forms (synonyms, labels) that syntactically identify concepts. Then, tokens from the input string are matched to this dictionary entries.

Second step is performed by *is_a transitive clo-*

sure, which aims to explore the relations in ontologies, namely for a given matched concept it searches its subsequent ancestors in the parent-child hierarchy and can match them as tags for a given token as well. The number of ancestors to look through is paramterizable.

Next, an *ontology-mapping component* tries to find relations between different ontologies, e.g., when a given concept is matched for a token, it can be linked to a respective concept in another ontology, and the ontology can also be traversed.

As a result, NCBO produces quite a lot of tags for each input text token. Our first trials showed that they are usually relevant, however often to many of them is generated. Hence the need to select the most suitable tag and possibly perform disambiguation.

6.2.2 Word2Vec

In order to work with word data, we have to represent it as numbers, preferably vectors. One of the encoding methods is one-hot encoding. This method allows us to change words to sparse vectors. There are many issues with this approach, for example:

- the distance between vectors is always the same,
- cosine similarity between vectors is always zero.

It is common to use Word2vec to overcome those issues (presented in (Mikolov et al., 2013)). This mechanism allows us to link word literals to dense vectors. Those vectors have valuable properties, such as meaningful usage of cosine metric. It is worth mentioning that if we take the embedding of the word "king" and subtract from that embedding of the word "man", and then we add the embedding of the word "woman" - we shall get the embedding of the word "queen". The concept is based on a simple neural network with the following architecture:

1. First layer with weights' matrix of size VxN with linear activation (projection layer).
2. Output layer with weights' matrix of size NxV with softmax activation.

Where: N is a parameter - the size of the hidden layer (also the size of embedding), and V is the size of the vocabulary. We can train this architecture with two approaches:

- Continuous Bag of Words (CBOW) - we want to train the neural network to predict the target word given its neighboring words. Projections from those words are averaged.
- Continuous Skip-gram - we want to train the neural network to predict neighboring words. The output of the second layer and the corresponding error are calculated separately for every neighboring word.

Those two approaches are illustrated on Figure 1.

The number of neighboring words is a parameter. When we choose this number to be equal to 5, we will choose five words from the future and the past i. e. ten neighboring words.

Note: In this description, we focused on the high-level idea behind Word2Vec, not on specific implementation issues or optimization techniques.

6.2.3 Words embedding tagger

The approach mentioned in (Karadeniz and Özgür, 2019) is based on the assumption that semantically similar words have similar vectors in the embedded space.

Before computing the word embedding vectors, the preprocessing is performed. The words need to be free from stop words, and non-ASCII characters.

Next, the word vectors are calculated, using the pre-trained model. For multi-word entities, each word is transformed separately and the average vector is calculated. After the conversion of both tagged data and ontology concepts, their similarity is measured. The authors proposed 2 similarity measurements. The cosine similarity is calculated by the given equation:

$$cosine\_similarity = \frac{AB}{\parallel A \parallel \parallel B \parallel}, \qquad (3)$$

where A and B are the vectors. The second metric is the word mover's distance (WMD). WMD treats tags as a weighted point in the cloud of the embedded words and calculates the minimum distance it needs to travel to a given concept. After performing those steps, the list of closest ontologies concepts is given.

6.3 Words tags disambiguation

In free-text data, the same word can occur in different contexts and with different meanings. For

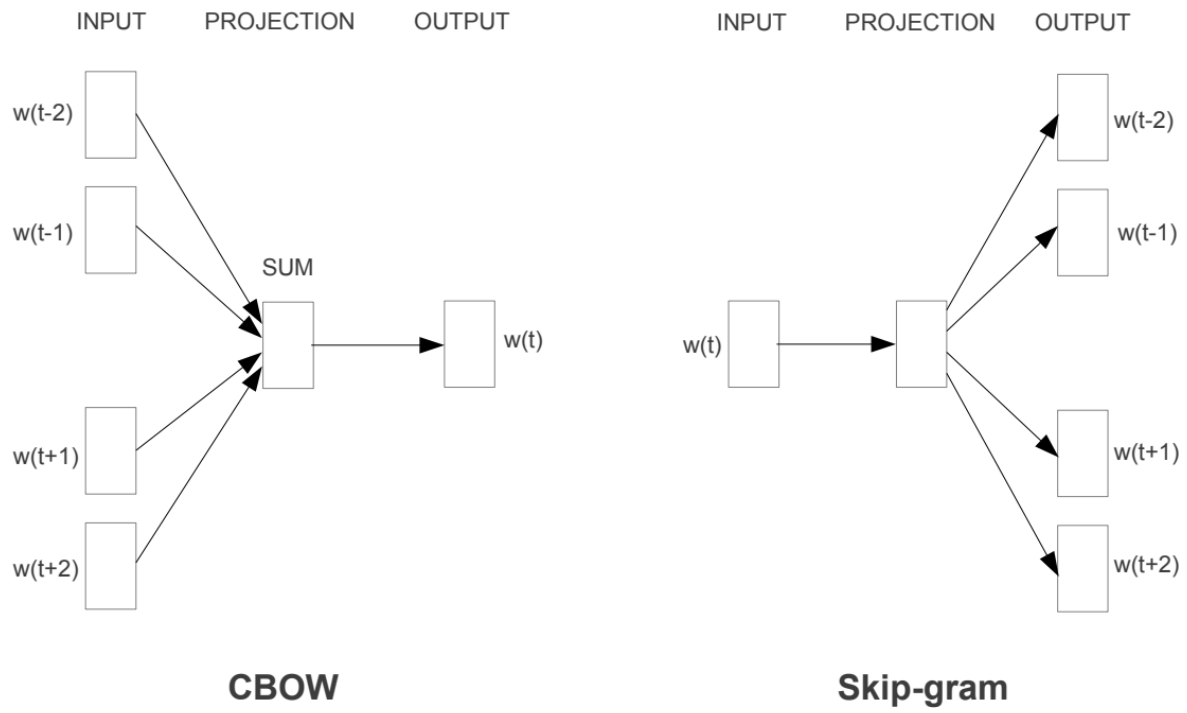

Figure 1: Illustration of CBOW and Skip-gram models (Mikolov et al., 2013).

example, if working with data containing information about wines, *Burgundy* can refer to the name of the wine or the region in France (Bindelli et al., 2008b). Hence, it should be decided whether to tag *Burgundy* with *Wine name* or *Country Region*. This information should be based on the context of a tagging word in an input text. If it comes to medical data, the term *blood pressure* can have three senses, namely *organism function*, *diagnostic procedure* and *laboratory or test result* (Alexopoulou et al., 2009).

We propose a method inspired by Alexopoulou et al. (2009). It is based on selecting the sense of a given word (or in general token) that is the closest to senses of other words appearing in the context. In the following subsections we describe the original Closest Sense method (sentence-based) and the modification than can be incorporated in our solution.

6.3.1 Sentence-based Closest Sense method

Let us suppose that we want to tag tokens in the sentence: *I also tracked lipid profiles, HBA1C, blood pressure, body mass index, hostility and nicotine use*. As mentioned above, *blood pressure* can have multiple senses since it is ambiguous - three tags are possible (assuming they are concepts of some ontology): *organism function*, *diagnostic procedure* and *laboratory or test result*.

To decide which tag should be assigned to *blood pressure*, we explore tags of other words appearing in the sentence. Let us assume that the senses of the occurring terms are *laboratory procedure* (lipid profile), *gene or genome* (HBA1C), *diagnostic procedure* (body mass index), *mental process* (hostility) and *organic chemical* (nicotine). Then for blood pressure we choose the sense that is on average closer to the senses of the co-occurring terms than the other candidate senses.

What the *closeness* means can be treated in various ways. For example, semantic distances utilizing the ontologies (like subsumption distance or subtype-aware signature distance) can be used (Alexopoulou et al., 2009). The other way could be to incorporate words embedding and investigate cosine similarity.

6.3.2 Keywords-based Closest Sense method

Since our task aims to tag keywords instead of particular words in a free-text, we plan to modify the Closest Sense algorithm.

First of all, for a given ambiguous keyword, we are going to treat other keywords extracted for a given text as the context, instead of words that occur in the same sentence.

Secondly, in the case of our problem, each key-

word is ambiguous on a similar level (all keywords will have numerous candidate tags assigned). This is a difference in regards to what Alexopoulou et al. (2009) explored: they assumed only a given word in a sentence is ambiguous while other words have correctly assigned tags. That is why we propose the following iterative procedure: given a set K of keywords k_j, $j = 1, \ldots, |K|$ for a given document and sets $|T_j|$ of candidate tags: t_{j,i_j} for j-th kyeword, $i_j = 1, \ldots, |T_j|$ perform *max_iter* times:

1. Take subsequent keyword k_j and assume this keyword is ambiguous, while all other keywords have correct tags assigned (take first tag in the candidate list for these keywords).

2. For each candidate tag t_{j,i_j} calculate the similarity distances to other keywords tags and sort the list of tags from $|T_j|$ by decreasing similarity distance. As a result, at the top of the list we have the best tag for k_j according to the current state.

3. Go to point 1. taking next keyword.

After max_iter iterations of above points, each k_j will be considered max_iter times. This heuristic can help to choose keywords tags according to the context of the entire document.

We plan to use similarity metric based on words embedding, but other choices mentioned earlier are possible.

7 Methods of results analysis

To compare tagging methods, we will use two metrics: precision, recall, and F1 score, which are defined as follows:

$$precision = \frac{TP}{TP + FP} \tag{4}$$

$$recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \tag{6}$$

Where:

- True positives (TP) - number of cases when our annotation matches annotation from golden standard dataset
- False postives (FP) - number of cases when our annotation doesn't match annotation from golden standard dataset
- False negatives (FN) - number of cases when annotation from golden standard dataset is not present in our annotations

It is hard to come up with a way to calculate the number of true negatives. That is why we choose metrics that do not rely on this particular number.

References

Dimitra Alexopoulou, Bill Andreopoulos, Heiko Dietze, Andreas Doms, Fabien Gandon, Jörg Hakenberg, Khaled Khelif, Michael Schroeder, and Thomas Wächter. 2009. Biomedical word sense disambiguation with ontologies and metadata: automation meets accuracy. *BMC Bioinformatics*, 10(1):28, Jan.

Silvia Bindelli, Claudio Criscione, Carlo Curino, Mauro Drago, Davide Eynard, and Giorgio Orsi. 2008a. Improving search and navigation by combining ontologies and social tags. pages 76–85, 11.

Silvia Bindelli, Claudio Criscione, Carlo Curino, Mauro Drago, Davide Eynard, and Giorgio Orsi. 2008b. Improving search and navigation by combining ontologies and social tags. pages 76–85, 11.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, mar.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl$_1$) : $D267 - -D270, 01$.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning, URL:* `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html`.

Maarten Grootendorst. 2022a. Bertopic - pypi, url: `https://pypi.org/project/bertopic/`.

Maarten Grootendorst. 2022b. Bertopic: Neural topic modeling with a class-based tf-idf procedure.

Clement Jonquet, Nigam Shah, Cherie Youn, Mark Musen, Chris Callendar, and Margaret-Anne Storey. 2009. Ncbo annotator: Semantic annotation of biomedical data. *ISWC*, 01.

Clement Jonquet, Nigam Shah, Cherie Youn, Mark Musen, Chris Callendar, and Margaret-Anne Storey.

2021. Ncbo annotator rest api, url: https://bioportal.bioontology.org/annotator, 01.

İlknur Karadeniz and Arzucan Özgür. 2019. Linking entities through an ontology using word embeddings and syntactic re-ranking. *BMC Bioinformatics*, 20(1):156, Mar.

Robert Leaman and Zhiyong Lu. 2016. TaggerOne: joint named entity recognition and normalization with semi-Markov Models. *Bioinformatics*, 32(18):2839–2846, 06.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11), mar.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Sunil Mohan and Donghui Li. 2019. Medmentions: A large biomedical corpus annotated with umls concepts.

Manuel Möller, Michael Sintek, Ralf Biedert, Patrick Ernst, Andreas Dengel, and Daniel Sonntag. 2010. Representing the international classification of diseases version 10 in owl. In Joaquim Filipe and Jan L. G. Dietz, editors, *KEOD*, pages 50–59. SciTePress.

Claire Nédellec, Robert Bossy, Estelle Chaix, and Louise Deléger. 2018. *Text-mining and ontologies: new approaches to knowledge discovery of microbial diversity*. May.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, August. Association for Computational Linguistics.

Radim Rehurek. 2022. Gensim - pypi, url: https://pypi.org/project/gensim/.

Nils Reimers and Iryna Gurevych. 2019a. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Astghik Sargsyan, Alpha Tom Kodamullil, Shounak Baksi, Johannes Darms, Sumit Madan, Stephan Gebel, Oliver Keminer, Geena Mariya Jose, Helena Balabin, Lauren Nicole DeLong, Manfred Kohler, Marc Jacobs, and Martin Hofmann-Apitius. 2020. The COVID-19 Ontology. *Bioinformatics*, 36(24):5703–5705, 12.