EPFL Taskbase

A Comparative Study of Task Modelling Techniques in the Field of Digital Education

by **Faten Ghali**

A thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Science in Data Science

École Polytechnique Fédérale de Lausanne
School of Computer and Communication Sciences

Thesis Advisor:
Prof. Martin Jaggi

Thesis Supervisor:
Luca Rizzello

June 20, 2025

Abstract

With the increasing demand for digital learning and the advancement of machine learning techniques, automatic scoring methods keep evolving. Providing a formative feedback relies on determining the different aspects to assess in the student's answer. This work suggests five different solutions to address the problem of automatically recommending the set of competencies to be evaluated in a given task. The solutions rely on topic modelling techniques, multi-label text classification (MLC), approximate nearest neighbor (ANN) on BERT word-embeddings and generative AI using GPT. The text-search method based on BERT and ANN outperforms the other solutions suggesting aspects that are similar to reality. The topic detection techniques are victim of the sparseness of short texts and MLC is beated by the large number of labels in a small dataset. The GPT-based solution provides interesting results at the cost of prompt engineering and post-processing. This work represents a starting point to develop sophisticated systems to automate the generation of feedback on E-learning platforms.

Acknowledgement

This work could not have been accomplished successfully without the help of many people.

A special thank you to Luca Rizzello, my supervisor at Taskbase, whose guidance, insightful feedback and unwavering support have significantly shaped the development of this work.

To Prof. Jaggi, my academical advisor, thank you for your expertise and for giving me valuable input and suggestions to improve this content.

To the task modelling team and everyone else at Taskbase who helped me during this internship.

To Emyna, Lilia, Simon and my close friends for the mental support and the nice memories.

To Adel, Rym, Sana, Mohamed and my parents, thank you for everything.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last years, with the adaptation to new forms of routine, including the openness to remote working and homeschooling, the need for access to digital learning increased significantly. With the expansion of online education platforms, the necessity to improve the automatic scoring methods emerged. Studies show that tailored formative feedback, as opposed to merely assigning a grade, is important to help students learn better, understand their progress and improve their engagement [1]. In recent literature, it is stated that all systems of automatic short answer grading are based on measuring the similarity between the student answer and a model answer to generate a final score [2].

Aware of the need for personalized feedback, Taskbase uses cutting-edge techniques not only to improve the learner's journey, but also to facilitate the creation of educational material for teachers and E-learning platforms, hereafter the tenants. While existing systems rely on a reference answer to evaluate the student's answer, Taskbase uses the concept of aspects to evaluate the task on different levels defined by the tenants.

1.1 Taskbase

Taskbase is a Swiss startup that specializes in the field of digital education. Its mission is to increase the success of learners and to support teachers in digital learning environments by evaluating tasks based on didactic concepts and mapping unstructured learner answers to competency-based information. The Aspect Engine, which is Taskbase's core product, is the foundation for building adaptive, personalized learning paths and high-quality content in learning products in a sustainable way. This product relies on two main concepts: tasks and aspects.

Tasks. A task is an exercise created by the tenant, to assess one or multiple

competences of the student. It is defined by a unique ID, a title, a type and a description. The description is the wording of the task as it is received by the student. We will only focus on text tasks, particularly those that demand an open answer.

Aspects. An aspect specifies a competence that would be evaluated during the task assessment. Aspects are essentially labels assigned to potential feedback. For example, consider a task description like: *"Explain what you did last week-end using the verb 'to do' in the past simple tense"*. Multiple aspects can be valuable in evaluating the student's response to this task, such as:
- An aspect evaluating whether the student is capable of using the past simple tense.
- An aspect evaluating whether the student is capable of using the required verb.
- An aspect to check if the student makes spelling mistakes.

An aspect is defined by a unique ID, a name, a description, a category and a type that can either be a "CONCEPT" or a "MISCONCEPTION". A CONCEPT aspect states a capability of the student, or what they would do correctly in their answer; a MISCONCEPTION aspect states a possible mistake. For instance, in the example above, one of the CONCEPT aspects could be *"The student is able to conjugate the verb 'to do' in the past tense"* and one of the MISCONCEPTION aspects *"The students did not use the required verb in the sentence"*.

1.2 Problem statement

Up until now, the tenants create manually and individually the aspects to evaluate the student's answer. Recommending proper aspects automatically would accelerate the process of task creation, automate the generation of the detectors[1], prevent the duplication of aspects, and improve the quality of the dataset.

This work suggests five different solutions to the problem of the aspects' recommendation and compares them. Exploratory data analysis, data cleaning, as well as pre-processing steps are presented in Chapter 2. Chapter 3 outlines the first solutions that are based on topic detection techniques: Latent Dirichlet Allocation and Latent Semantic Analysis. In Chapter 4, we opted for machine learning and generative AI: Section 4.1 implements multi-label text classification using a BERT-based model, Section 4.2 presents the results of a search-engine-like solution using BERT-tiny for embeddings and approximate nearest neighbor for recommendation, and Section 4.3 relies on GPT for recommending descriptions of aspects for input tasks. Chapter 5 presents general discussion of the solutions and future work.

[1] Detectors are algorithms implemented specifically to evaluate the aspects and provide the feedback. The creation of the detectors follows the creation of the aspects.

Chapter 2

Exploratory Data Analysis and Pre-processing

2.1 Exploratory data analysis

Tasks. There exist 32 possible types of tasks as shown in Figure 2.1. Our focus will be on the OPEN type as it is the third most used type and representing 15% of the dataset. The two task types more common than OPEN are either too simple, having repetitive aspects and description (i.e, multiple choice tasks) or too hard, requiring a mapping of multiple aspects per gaps (i.e, cloze text tasks).

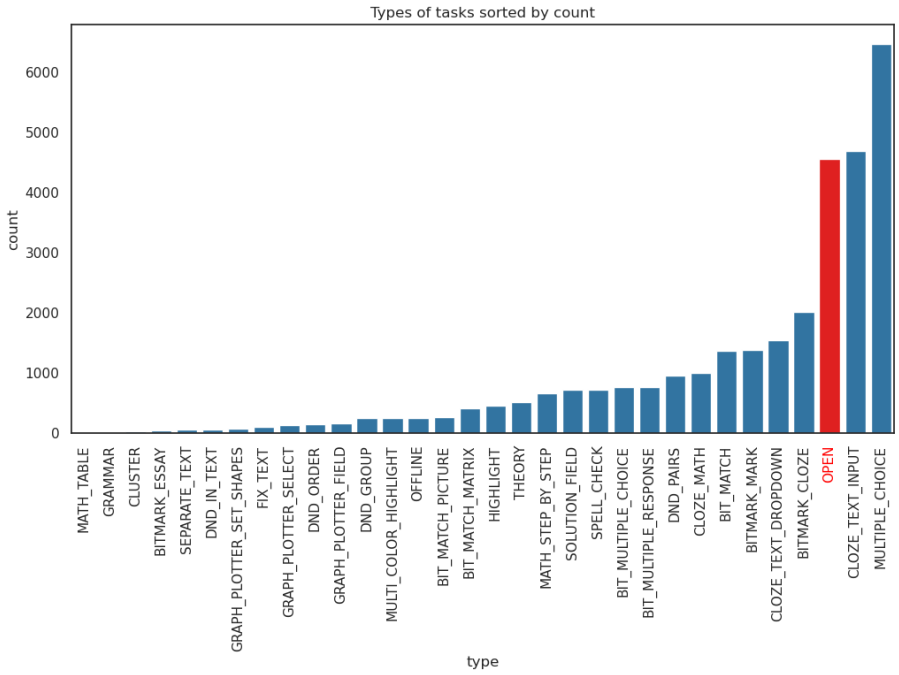


Figure 2.1: Barplot of the number of tasks per type showing the important usage of the OPEN type.

Figure 2.2 shows that almost all tenants rely on type OPEN for their tasks and a fair number of them use it exclusively, which stresses the importance of this work to automate the generation of aspects related to this type of tasks.

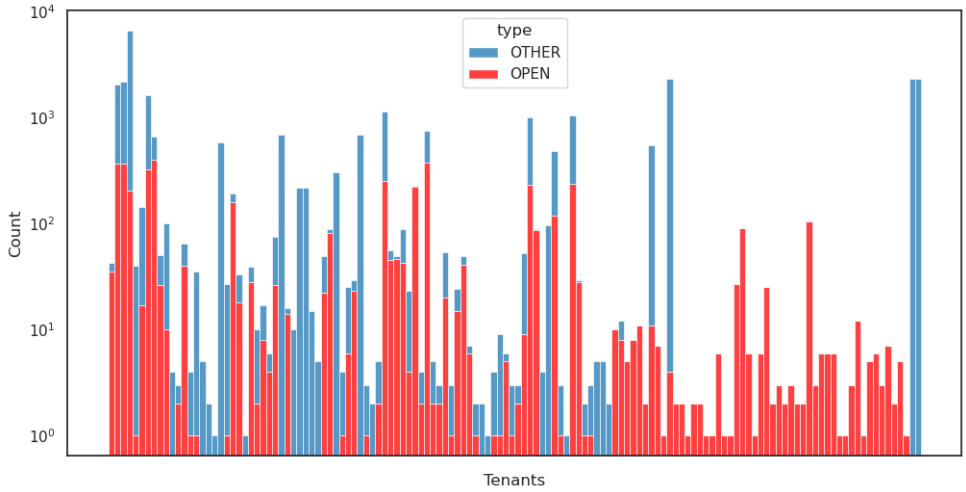


Figure 2.2: Types of tasks by tenant.

Figure 2.3 shows the distribution of the four languages present in the dataset and justifies the focus on the German and English tasks in the scope of this thesis. German and English tasks represent 67% of the dataset, while "NaN" represents the category with unspecified language and accounts for 26%.

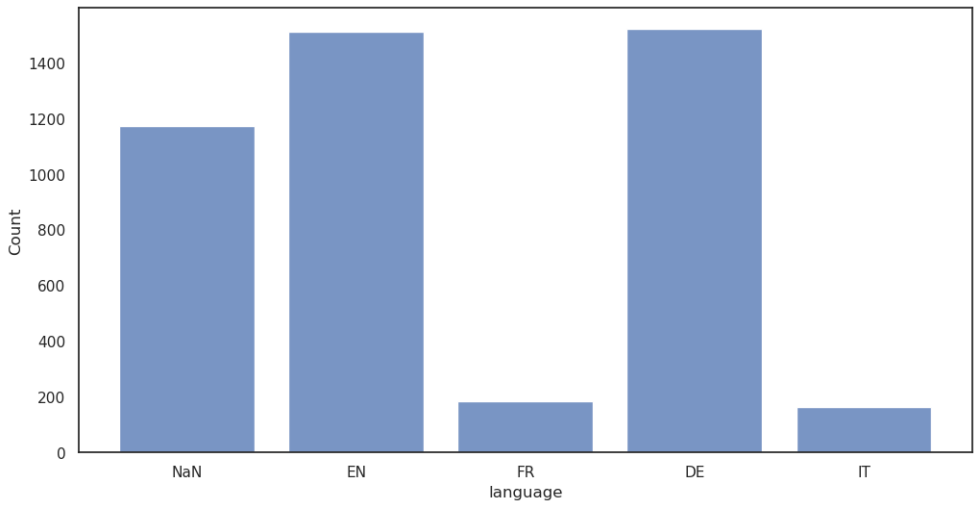


Figure 2.3: Number of OPEN tasks per language.

The "topic" field of OPEN tasks is sparse −only 27% of the tasks have it specified. Hence, it is not reliable for classifying the tasks or providing more information about their content.

To gain deeper insight into the German and English tasks, we compute the lengths of their descriptions in words and plot them as in Figure 2.4. The distribution is heavy-tailed and the log scale on the x-axis allows us to see the presence of short or

empty descriptions. Indeed, empty tasks account for 6%, and those of less than 5 words represent 7% of the German-English dataset. The descriptions with less than 5 words might be testing tasks (like "test question", "this is a test"...) and can be discarded. Those with less than 10 words however represent 23% of the set and are more likely to be real tasks. The tail of the distribution shows that less than 1% of the German and English tasks are at least 100 words long.

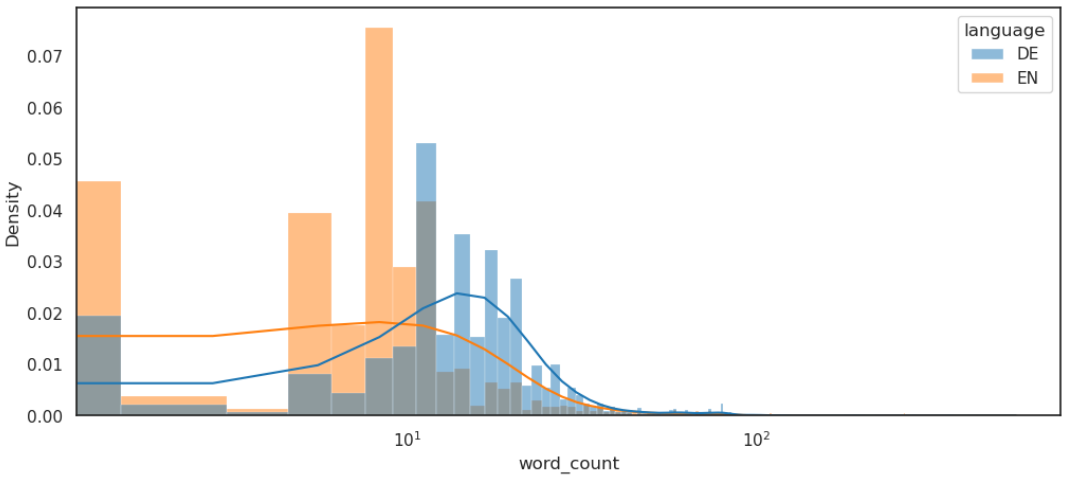


Figure 2.4: Distribution of descriptions' lengths in words.

Aspects. In order to know how many aspects can be assigned to a task, we plot the distribution of the number of aspects per task as shown in Figure 2.5. The figure does not show a particular statistical distribution and the number of aspects has a quite large range ([1,105]), but most importantly the figure shows that almost 50% of the tasks in the database do not have any assigned aspects. These could be created for testing purposes only, but they are not easily detected because most of them have a proper description and are assigned to an existent tenant. Thus, these tasks will be discarded in the cleaning process.

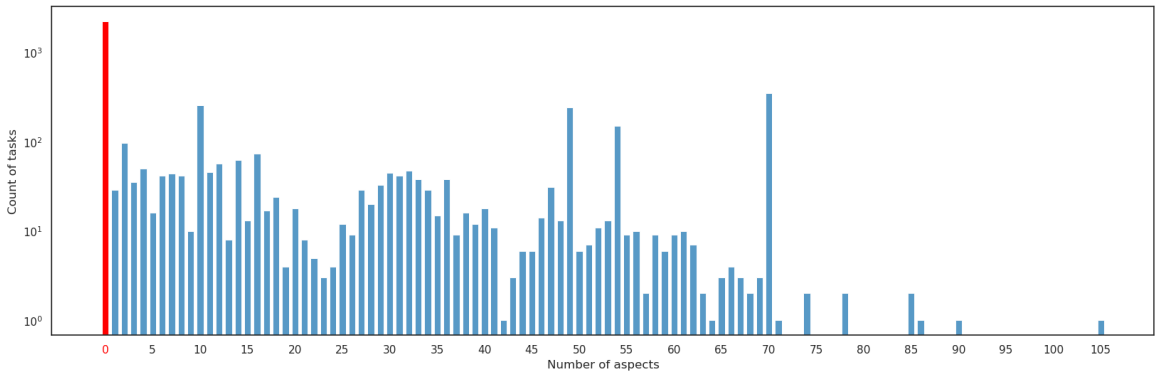


Figure 2.5: Histogram of number of aspects per task.

Upon closer examination of the assigned fields, aspects come in categories and groups. Figure 2.6 shows the unbalanced distribution of categories and that 44%

of the German-English set of aspects does not have a specified category ("NaN"). Groups, on the other hand, are too specific: for 1853 aspects with a specified group,

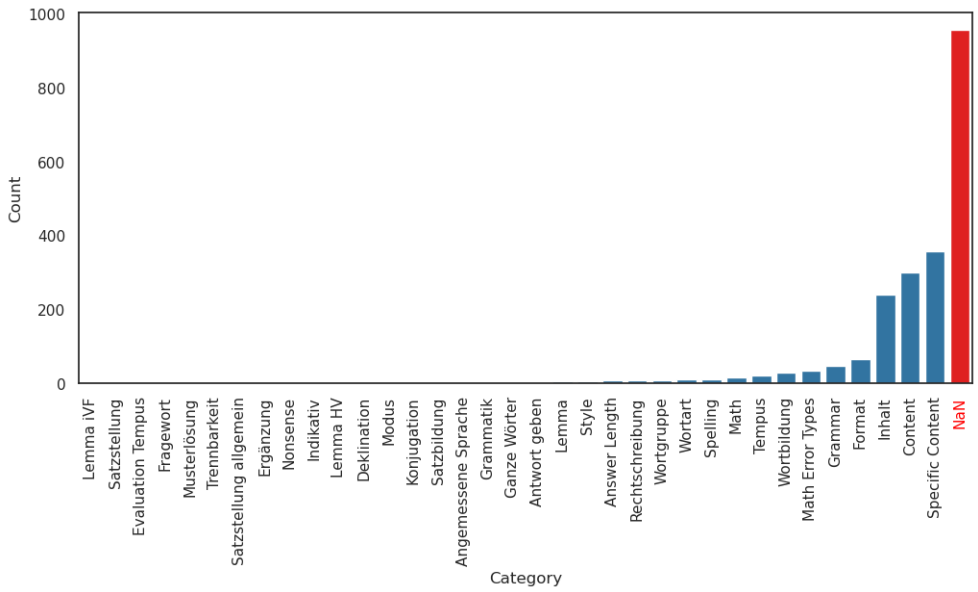


Figure 2.6: Distribution of German and English aspects' categories.

we count 1562 unique groups. Hence, the category and group properties would not be reliable criteria for classification of aspects.

2.2 Data cleaning

Tasks. As a first step of cleaning the tasks, we remove the ones with unspecified tenant and those with a "Playground" (dummy) tenant because the tasks' quality and their mapping to relevant aspects are not guaranteed in these cases. The removed tasks account for about 14% of the dataset. After that, we remove the tasks of short descriptions (less than 5 words) for the reasons explained above. Next, we use the NLTK library to detect the language of tasks when it is unspecified. 546 German and English tasks were recovered after this step.

At the end of these cleaning steps, the dataset contains 730 German and 1161 English tasks and the distribution of their descriptions' lengths is depicted in Figure 2.7. We leave the decision of removal or truncation of long descriptions to later sections depending on the needs of the project.

At the end of the cleaning process, we make a 90-10 split to create training and testing sets.

Aspects. In the scope of this work, we will only consider aspects of type "CONCEPT" for a couple of reasons: First, every task has to have at least one

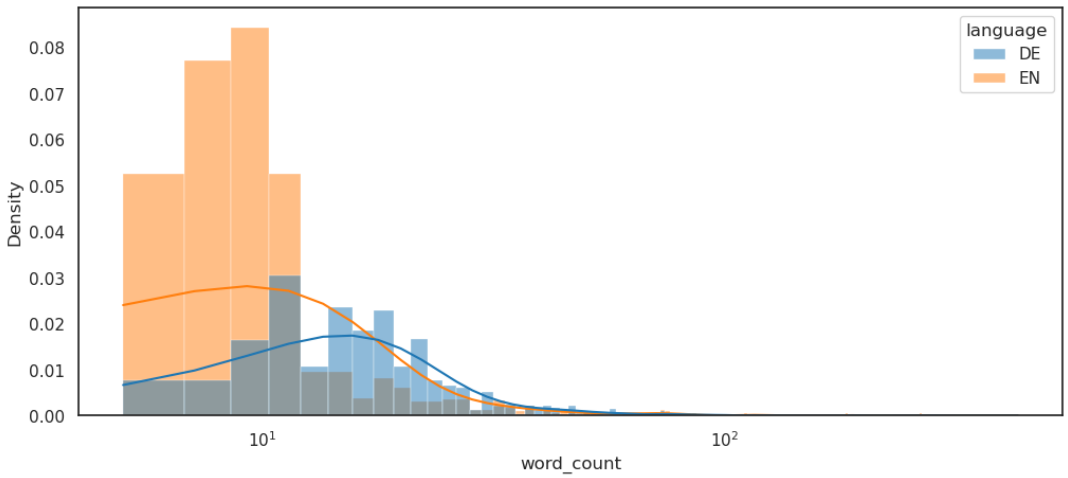


Figure 2.7: Distribution of descriptions' lengths in words.

CONCEPT aspect, and second, the MISCONCEPTION aspects can then be directly derived from the CONCEPT ones. Figure 2.8 shows the distribution of the number of CONCEPT aspects per task in the German and English OPEN tasks after the cleaning steps. We observe that the English tasks tend to have a larger number of aspects attributed compared to the German ones. In fact, the average number of CONCEPT aspects is 22 for a German task and 48 for an English one.

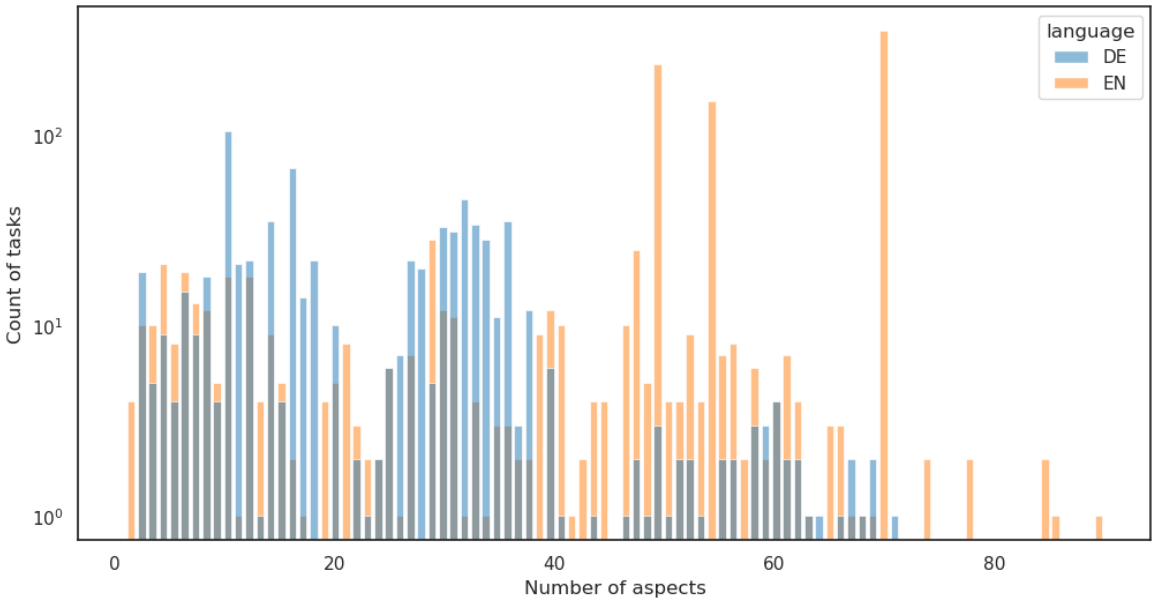


Figure 2.8: Histogram of number of aspects per task after cleaning.

2.3 Data augmentation

In order to have a richer and more diverse training dataset we proceed to augmentation, which is proven to improve the performance of language models. EDA, for Easy Data Augmentation, is a tool made available[1] to make easy the implementation of simple text augmentation. These techniques boost performance on text classification tasks and demonstrate particularly strong results for smaller datasets like ours[3]. The techniques used here except for back translation were implemented using EDA:

- **Back translation (BT)** consists of translating English text to German and back to English, and German text to English and then back to German. This operation often results in a new sentence that is slightly different from the original with preservation of its meaning.
  Example: "Write a phrase in English" -> "Schreiben Sie einen Satz auf Englisch" -> "Write a sentence in English"

- **Synonym replacement (SR)** randomly selects two tokens from the text and replaces them by synonyms from the NLTK dictionary which should preserve the meaning of the original text.
  *Example: "Write a **phrase** in English" -> "Write a **sentence** in English"*

- **Random insertion (RI)** randomly selects two positions in the input text to randomly insert a new token in each.
  *Example: "Write a phrase in English" -> "Write a **long** phrase in English **language**"*

- **Random swap (RS)** randomly selects two tokens form the input text and swap them resulting in a new order of words in the text.
  *Example: "Write a phrase in English" -> "**phrase** a **Write** in English"*

- **Random deletion (RD)** deletes tokens from the input text with a given probability $p = 0.4$.
  Example: "Write a phrase in English" -> "Write a in English"

All techniques are separately applied to each training set and the results are then concatenated to it, resulting in the end in a dataset comprised of 3942 German and 6270 English descriptions, i.e. 6 times the size of the original.

[1] http://github.com/jasonwei20/eda_nlp

	Original	After cleaning		After augmentation	
		Train	Test	Train	Test[2]
German	965	657	73	3942	73
English	1512	1045	116	6270	116

Table 2.1: Count of OPEN tasks before and after data cleaning and augmentation.

2.4 Data pre-processing

After concatenating the augmented data together, we proceed to data pre-processing. These steps are common in Natural Language Processing (NLP) to clean the input text and are applied before any processing. First, we change the text to lower case, we then remove all punctuation. After that, we remove the stopwords, which are the common words that do not belong to a specific context: *I, me, my, you, your, yours, it, he, she....* Finally, we apply stemming and lemmatization as processes to reduce the variant word forms to their base forms. Stemming reduces the words to a stem merely by abridging the suffix to its base root and the semantic meaning remains the same. For example, "played" and "playful" both reduce to "play", but "run" and "ran" remain unchanged. Lemmatization, however, removes the inflectional endings and returns the base or dictionary form of the word to reduce its variations. For example, the words "running", "run", "ran" will all return to the base form "run" that we call a lemma [4].

[2]not augmented

Chapter 3

Topic Detection

Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA) are topic modelling algorithms. They are widely used to detect topics in text corpora and easily fit into various applications. This chapter will explore the use of these techniques to recommend content that aligns with the themes detected in the input text.

LDA was first introduced by Blei et al. in [5] as a generative probabilistic model for collections of discrete data such as text corpora where each document is a finite mixture of topics. Topics, in turn, are each modeled as a finite mixture of an underlying set of topic probabilities. The generated probabilistic topics of LDA help make the texts more semantic-focused and reduce their sparseness. Chen et al. benefited from this feature to employ LDA with K-Nearest Neighbor algorithm in the context of short text classification [6].

LSA, on the other hand, identifies conceptual similarities instead of explicitly modelling the topics. It in fact relies on singular value decomposition to reduce the dimensionality of word-document matrices and captures latent semantic relationships between terms and documents [7]. LSA has proven capable of extracting meaning from passages of texts and have been used in various applications across different domains of linguistics, psychology, cognitive sciences and education [8].

For the evaluation of topic models, coherence scores such as C_V, U_Mass, and C_UCI are commonly used. The C_V score measures coherence based on a sliding window, using a combination of indirect confirmation measures and cosine similarity. The U_Mass score, derived from document co-occurrence probabilities, is simpler but often less aligned with human judgment. Lastly, C_UCI evaluates coherence based on pointwise mutual information between words, which captures the strength of association between terms.

This chapter explores different ways of employing LDA and LSA to solve the aspects recommendation problem.

3.1 Latent Dirichlet Allocation (LDA)

3.1.1 Methodology

The method relies on LDA to cluster the tasks based on the topics of their descriptions. As a first step, we run different LDA models with different sets of parameters and report the coherence score of each. The U_Mass coherence score was chosen because it yielded better results compared to the C_V and the C_UCI scores.

The topics will be described by the top ten representative terms, i.e. the ten most frequent terms in the cluster. This should help form an idea about the topic's content.

In the second step of the solution, we explore the aspects of the tasks in each cluster and analyze their disparity in order to attribute a list of aspects to each topic. The final model would then decide the topic, or topics, of the input task and suggest the list of appropriate aspects accordingly.

Through the implementation of this solution, a number of problems arose. In the following sub-sections, we provide a detailed analysis of the different problems we encountered and the methods we applied to overcome them.

A. Baseline

The baseline solution consists of applying the aforedescribed method on the descriptions as they are presented after the cleaning, augmenting and pre-processing steps. We run the LDA model using multiple combinations of parameters; varying the number of topics (2, 3, 4, 5) and the number of passes made on each document (20, 40, 60, 80 or 100) which results in 20 models. The parameters *iterations* and *decay* are fixed and set to 100 and 0.8 respectively.

Table 3.1 reports the coherence scores of the 20 models run on the German (on the left) and the English documents (on the right) separately. It shows that the model (topics=2, passes=80) outperforms the others on the German corpus. For the English data, the model shows more coherence when the number of topics is set to 2 with 100 passes on each task.

To take a closer look at the content of the clusters, the ten most representative terms for each cluster are listed below. These terms are actually lemmas, as this is the format of the model input.

- German:
  Topic 0: *satz, schreib, prasen, passiv, beginn, fur, markiert, schreiben, satzteil,*

German

# passes \ # topics	2	3	4	5
20	-7.60	-8.64	-9.56	-9.77
40	-7.18	-6.86	-8.80	-8.54
60	-7.68	-7.95	-10.08	-9.22
80	**-6.58**	-9.02	-8.96	-8.74
100	-7.15	-9.89	-10.15	-9.19

English

# passes \ # topics	2	3	4	5
20	-9.79	-9.94	-9.27	-10.61
40	-7.42	-10.30	-9.77	-9.56
60	-9.97	-7.99	-10.32	-10.78
80	-9.61	-9.01	-9.88	-10.42
100	**-7.21**	-7.81	-9.76	-10.27

Table 3.1: U_Mass coherence scores of the first run of LDA models on German and English tasks.

perfekt[1]

Topic 1: *satz, schreib, frage, fur, infinitiv, antwort, zwei, geben, beispiel, komma*[2]

- English:

  Topic 0: *write, sentenc, facil, use, one, word, answer, follow, like, question*

  Topic 1: *sie, satz, ubersetzen, translat, den, english, englisch, auf, schreib, horst*

Two major problems can be pinpointed:

1. German words appearing among the representative words of English topic 1.
2. Terms like "fur" and "zwei" in German or "one" and "use" in English are of wide use and can hardly be representative of one topic.

Problem 1 could be caused by the translation tasks. Typically:
Translate this sentence into English: "Tom wohnt in einem grossen Haus".
To overcome this problem, we will translate all German tasks (again) to German, and English tasks to English, which should keep the tasks with a unique language as intact as possible.

Problem 2 arises from the fact that LDA relies on the occurrences of terms in the text. The adopted solution here is enlarging the list of stop-words with common words like "one" and "use" in the pre-processing steps.

B. Second round of data pre-processing

To preserve the context of the tasks, we perform the suggested solutions on the original dataset before any pre-processing. We first apply the translation to unify the language. Then, we proceed to the usual augmenting and pre-processing steps. We eliminate the terms of unspecific topic ("say", "like"...) after the lemmatization step to make it easier. Figure 3.1 shows the heavy-tailed distribution of the lemmas in both languages, so it is sufficient to view the top 50 lemmas and manually pick

[1] sentence, write, present, passive, begin, for, marked, writing, clause, perfect
[2] sentence, write, question, for, infinitive, answer, two, give, example, comma

the ones to remove. We picked {*fur, zwei, of, gehen, bei, perfekt, bitt, wichtig, gut, konn, uber, drei*[3]} for German and {*use, one, make, tom, like, take, get, say, go, want, two, could*} for English.

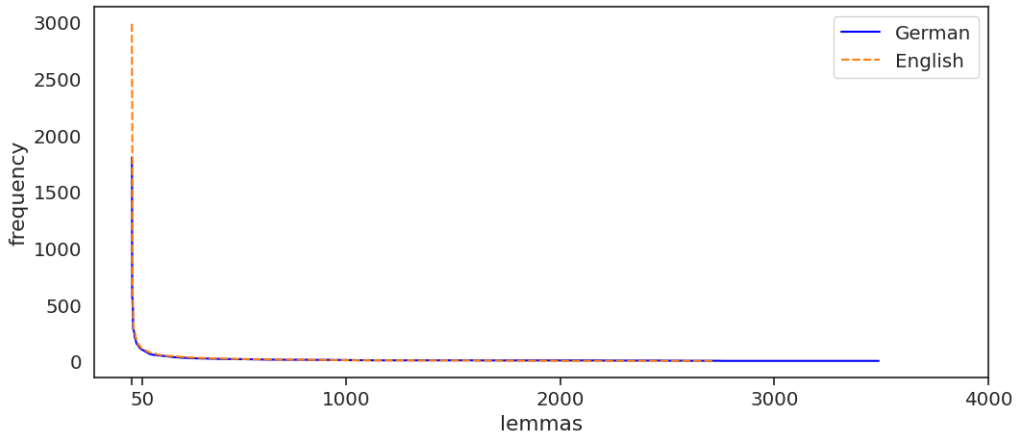


Figure 3.1: Distribution of lemmas' frequencies in clean augmented descriptions.

After applying the solutions suggested in the previous sub-section, we run LDA on the obtained data and select the model with the highest coherence score as previously. As shown in Table 3.2, the model (topics=2, passes=60) works best on the German data, keeping the same number of topics as before; while the English data scores a better coherence with the model (topics=3, passes=80).

German

# passes \ # topics	2	3	4	5
20	-10.63	-9.05	-10.19	-11.99
40	-8.38	-9.81	-10.86	-10.41
60	**-7.84**	-8.70	-10.55	-12.11
80	-8.47	-8.68	-10.57	-11.75
100	-9.94	-9.36	-11.06	-11.72

English

# passes \ # topics	2	3	4	5
20	-9.62	-12.09	-9.98	-10.29
40	-10.03	-10.05	-10.33	-10.53
60	-9.37	-9.42	-10.84	-11.37
80	-10.46	**-8.47**	-11.90	-11.21
100	-9.92	-11.45	-9.86	-10.90

Table 3.2: U_Mass coherence scores of LDA models after the second round of pre-processing of the tasks.

When looking at the representative terms, there is no apparent improvement in the detection of the German topics. The English topics, however, are slightly more distinguishable than before despite the degradation of the coherence score: Topic 0 seems to be mostly about listening exercises, and Topic 1 about translation tasks, but it is still hard to conclude the subject of Topic 2.

- German:
  Topic 0: *schreib, satz, infinitiv, nenn, beispiel, satzen, geben, antwort, haupt-satz, komma*[4]

[3]for, two, off, go, at, perfect, please, important, good, can, about, three
[4]write, sentence, infinitive, name, example, sentences, give, answer, main clause, comma

Topic 1: *satz, schreib, prasen, passiv, frage, markiert, schreiben, satzteil, beginn, prateritum*[5]

- English:
  Topic 0: *write, hear, english, answer, name, question, text, explain, number, tree*
  Topic 1: *translat, sentenc, english, follow, write, word, comput, sit, front, read*
  Topic 2: *edinburgh, sentenc, work, write, citi, studi, day, friend, father, carolin*

For a given document, the LDA model attributes topics with a likelihood each, that together sum up to 1. We decided to discard the topics with a probability less than $1/n$, n being the number of topics, for all the tasks before attributing labels.

3.1.2 Labels attribution

After the topic detection of the tasks' descriptions, we attribute a collection of aspects for each topic by concatenating the aspects of tasks in each cluster. For example, if task A has aspects 1 and 2, and belongs to topic T, then we attribute aspects 1 and 2 to topic T.

The main downsides of this solution include the potential that it can create a large list of aspects for each topic, and the possibility that some aspects can be inaccurate for certain topics. For example, a task a has topics T and U, and aspects 1 and 2. Task a is attributed aspect 1 because of topic T and aspect 2 because of topic U. Despite the fact that aspect 1 represents better topic T and aspect 2 topic U, both topics will have both aspects attributed. So, if a new task b has topic T detected, this solution would recommend both aspects 1 and 2 instead of only aspect 1 which would be more accurate. As an attempt to make the attribution of aspects to topics more accurate, we remove the least frequent aspects in each topic. To do so, we define a threshold of occurrence t based on the values of the occurrences in each topic, as opposed to a fixed value which risks having empty sets. t is defined by the median value of the aspects' occurrences in the corresponding cluster. Table 3.3 compares the number of aspects before and after this reduction. The German aspects are reduced by 74% and the English aspects by 60% on average.

3.1.3 Testing and evaluation

To test the solution on a new task, we apply the pre-processing steps on the input (translation, removal of stopwords, lemmatization and removal of additional stop-lemmas), then we fit it to the pre-trained model. The model outputs the detected

[5]sentence, write, present, passive, question, marked, write, sentence part, beginning, past tense

	Topic	t	#labels before	#labels after
German	0	6	574	107
	1	6	457	150
English	0	6	544	222
	1	6	520	239
	2	6	410	126

Table 3.3: Number of labels per topic before and after reduction based on threshold t.

topics along with their likelihoods. We discard the ones with $p < 1/n$, where n is the number of topics, and proceed to the attribution of labels. We simply collect the IDs of aspects previously mapped with the recommended topics. Table 3.4 shows test examples in both languages. The test examples show poor results of the model, suggesting the same topic for tasks of clearly different contexts.

To gain more insight on the performance of the model, we run it on the test set and evaluate the similarity of the suggested set to the true set of aspects using the Jaccard similarity score defined as:

$$Jaccard\_similarity(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where A and B are sets of aspects' IDs. The similarity score ranges from 0 to 1, where 0 means that the sets are completely disjoint and 1 means they are identical. The Jaccard similarity score between the suggested and true sets of aspects does not exceed 0.1 on average, and is equal to 0 in at least 15% of the cases, in both languages, confirming the poor results of this model.

Task	**Topic(s) ID(s)**	**Nb. sugg. aspects**	**Nb. true aspects**
German			
Nenne das Formelzeichen und die Einheit des elektrischen Widerstands.[6]	0	107	4
Übersetzen Sie den Satz: "Er hat ein Abendessen bestellt."[7]	0	107	8
English			
Translate "Hast du schon ein Bad genommen, Takashi?" to English.	[0,1]	461	8
How should one behave in a building during an earthquake?	0	222	1

Table 3.4: LDA models' output on test examples

[6]Name the symbol and unit of electrical resistance.
[7]Translate the sentence: "He ordered dinner."

The total dissimilarity of the true and recommended aspects (score equal to 0) is directly linked to an erroneous topic detection of the task. For the rest of the cases, the low similarity score can be the result of the fact that more than 60% of the tasks have only one aspect assigned, which leads to comparing a set of one element to a set of hundreds of elements. Despite the attempts to reduce the number of the recommended aspects, the latter still exceeds the average number of true aspects by hundreds. Only 34% of the German tasks and 54% of the English ones have the true set of aspects completely contained in the set of recommended aspects.

3.2 Latent Semantic Analysis (LSA)

3.2.1 Methodology

As done previously, we run different LSA models and choose the one with the highest coherence score. This time, the C_V score is used. LSA is run on the texts of tasks' descriptions after a second round of translation to the original language for uniformity, lemmatization, and removal of certain "wide-use" lemmas.

As demonstrated in Section 3.1.2, attributing the labels of all the tasks combined in one topic cluster does not yield high quality results. Thus, as a way to recommend the aspects more efficiently, they are also clustered by topic using LSA. The performances of the LSA models run on tasks and aspects' descriptions are presented in Table 3.5, highlighting the best candidates.

German

Tasks		Aspects	
#topics	score	#topics	score
2	0.363	2	**0.507**
3	0.371	3	0.501
4	**0.379**	4	0.502
5	0.370	5	0.479

English

Tasks		Aspects	
#topics	score	#topics	score
2	0.327	2	0.405
3	0.345	3	0.404
4	0.415	4	0.441
5	**0.448**	5	**0.456**

Table 3.5: C_V coherence scores of LSA using different numbers of topics on tasks and aspects.

Unlike LDA, for a given input text, LSA outputs the distance to the different detected topics instead of probabilities. These distances are computed using cosine similarity between the lemmas of the input text and those of the top terms of the topic. As a result, for each text we assign one topic: the one with the shortest distance. This model outputs 4 task topics and 2 aspect topics for the German data. For the English data, 5 task topics and 5 aspect topics were detected. Table 3.6 shows some of the top terms in every topic. This visualization technique is

limiting, but it is still clear that topics tend to overlap (example of topics 1, 2, and 3 of the English tasks).

German

Tasks	0: ['satz', 'schreib', 'prasen', 'passiv', 'beginn', 'markiert', 'infinitiv'...] 1: ['infinitiv', 'frage', 'satzteil', 'technolog', 'institut', 'denk'...] 2: ['infinitiv', 'komma', 'denk', 'satz', 'passiv', 'prateritum', 'plan'...] 3: ['frage', 'direkt', 'indirekt', 'lie', 'passiv', 'infinitiv', 'schreiben'...]
Aspects	0: ['antwort', 'lernend', 'wurd', 'enthalt', 'geben', 'verb', 'erkennen'...] 1: ['wurd', 'erkennen', 'ublicherweis', 'postkart', 'frau', 'erwahnen'...]

English

Tasks	0: ['hear', 'write', 'english', 'translat', 'sentenc', 'follow', 'comput'...] 1: ['translat', 'sentenc', 'comput', 'front', 'follow', 'english', 'work'...] 2: ['translat', 'sentenc', 'work', 'eat', 'understand', 'appl', 'cook'...] 3: ['english', 'translat', 'line', 'need', 'protect', 'bank', 'work', 'find'...] 4: ['hear', 'translat', 'sit', 'front', 'comput', 'phrase', 'let', 'work'...]
Aspects	0: ['word', 'know', 'student', 'answer', 'write', 'correctli', 'sentenc'...] 1: ['text', 'abil', 'correctli', 'spell', 'write', 'languag', 'answer', 'verb'...] 2: ['abil', 'answer', 'languag', 'right', 'formul', 'correct', 'convert'...] 3: ['languag', 'convert', 'text', 'know', 'abil', 'right', 'word', 'speak'...] 4: ['subject', 'verb', 'match', 'correctli', 'abil', 'congruent', 'know'...]

Table 3.6: Top terms of the different topics as detected by LSA

3.2.2 Labels attribution

After detecting task topics and aspect topics, the following sections present two methods to map the former to the latter.

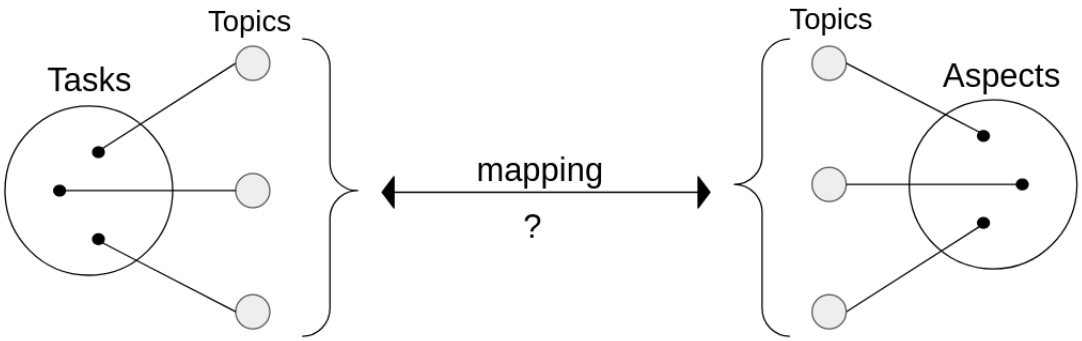


Figure 3.2: Illustration of the topics mapping problem.

A. First method

For the first method, we choose the most forward way of mapping the two sets of topics: collect the aspects assigned to each task in each task topic and map the different topics of those aspects to the topic of the tasks, as illustrated in Figure 3.3. For instance, given tasks a and b assigned a topic T, if a has aspects 1 and 2, and b aspects 3, 4 and 5, while aspects 1, 3 are assigned topic X and aspects 2, 4, 5 are assigned topic Y; then all tasks with detected topic T will be assigned aspects with detected topics X and Y.

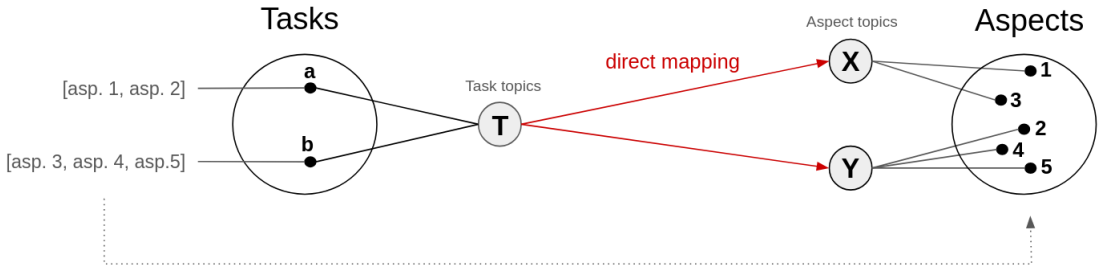


Figure 3.3: Illustration of the first method of mapping.

Table 3.7 shows the mapping results yielded by this method in number of topics and mapped aspects. In this case, the tasks from all the task topics had aspects from all the aspect topics which made all task topics being mapped to all the aspect topics, which is the worst scenario.

	Task topics	**Mapped aspect topics**	**# mapped aspects**
German	0	[0,1]	637
	1	[0,1]	369
	2	[0,1]	147
	3	[0,1]	441
English	0	[0,1,2,3,4]	533
	1	[0,1,2,3,4]	170
	2	[0,1,2,3,4]	542
	3	[0,1,2,3,4]	523
	4	[0,1,2,3,4]	86

Table 3.7: Results of first LSA mapping method

The first problem with this method is that tasks end up almost always being associated to hundreds of labels. The second problem is the reliability of this recommendation: how accurate are these labels?

B. Second method

To mitigate the problems encountered in the previous method −assigning the aspects of all topics to all the topics of tasks, we limit the assignment to the most probable aspect topics. To do so, we consider the previous setup and compute the probability for each task topic to be assigned any aspect topic. Given tasks of topic T, this probability of topic T being assigned to an aspect topic A is equal to the number of aspects of topic A over the total number of aspects assigned to the tasks of topic T. An example is illustrated in Figure 3.4. To prevent limiting the assignment of tasks to one topic of aspects and to solve the cases of equal probabilities, we use $t = 1/n$, where n is the total number of assigned aspect topics, to be the threshold for this probability. Details of these computations are shown in Appendix A.

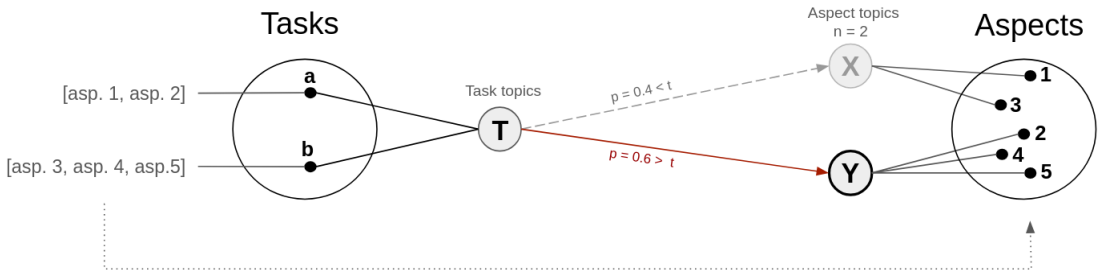


Figure 3.4: Illustration of the second method of mapping ($t = 1/n = 0.5$).

Table 3.8 compares the number of labels assigned to each task topic using the two methods of labels attribution. The second method reduces the number of aspects by 24.5% and 27.4% on average in the German and the English data respectively.

	Topic	#labels 1st method	#labels 2nd method
German	0	637	492
	1	369	279
	2	147	116
	3	441	304
English	0	533	397
	1	170	123
	2	542	373
	3	523	375
	4	86	63

Table 3.8: Number of labels per topic before and after reduction based on threshold t.

3.2.3 Testing and evaluation

Table 3.9 shows the output of LSA on a few example tasks and compares the number of suggested aspects to the number of true aspects.

Task	Topic(s) ID(s)	Nb. sugg. aspects	Nb. true aspects
German			
Nenne das Formelzeichen und die Einheit des elektrischen Widerstands.[6]	1	279	4
Übersetzen Sie den Satz: "Er hat ein Abendessen bestellt."[7]	0	397	8
English			
Translate "Hast du schon ein Bad genommen, Takashi?" to English.	3	375	8
How should one behave in a building during an earthquake?	2	373	1

Table 3.9: LSA models' output on test examples

The Jaccard similarity score is again used to quantify the similarity between the true and suggested sets of aspects. There is no occurrence of the suggested set being identical to the true set of aspects, the difference in the sizes of these sets as shown throughout this section makes it almost impossible to have perfect scores. The score is equal to 0 in 15% of the German tests and in 7% of the English ones. On average, the scores are not promising either: they do not exceed 0.01 in both languages. Despite the large sizes of the suggested sets, only 5% of the true sets were contained in the suggested. This might be explained by the fact that only two aspect topics were mapped. Aspects of topics 1 in German, and 1,3,4 in English were never suggested.

3.3 Conclusion

With both LDA and LSA, we observe the poor quality of topic detection: the topics are not distinct enough. This is in part related to the quality of the input texts. They are either too short or contain long text extracts that cause the deviation of the topic.

For the attribution of aspects, all the suggested solutions output a very large number of labels. A direct cause of this issue is the existence of a large number of aspects with little differences in the dataset: there exist aspects with very similar or identical descriptions and different IDs. Even though recommending aspects based

[6]Name the symbol and unit of electrical resistance.
[7]Translate the sentence: "He ordered dinner."

on similarity of tasks' topics is a plausible solution, the density and the imbalanced distribution of the aspects in the dataset makes it hard to be accurate in this context.

In conclusion, solutions based on LDA and LSA are not optimal for text recommendation.

Chapter 4

Machine Learning and LLMs

This chapter will rely on more advanced NLP techniques to recommend aspects to tasks. First leveraging BERT's (Bidirectional Encoder Representations from Transformers) capabilities of capturing rich contextual information from texts [9]. BERT, has been increasingly utilized in various NLP tasks like multi-label classification (MLC) of texts. MLC consists of assigning a given task multiple categories simultaneously. MLC was applied on BERT word-embeddings across various domains from legal to biomedical [10, 11].

Small variants of BERT have been extracted from the original larger model by pre-training and distillation. These models use less parameters to satisfy resource constraints while keeping a good performance on small tasks and datasets. BERT-tiny is a much smaller version of BERT composed of two layers instead of 12 and hidden embeddings of size 128 instead of 768 [12]. This model is used in Section 4.2 to compute the embeddings of tasks. Embedidngs being the primary component of our approximate nearest neighbor (ANN) search index. Instead of returning the closest point in a target space P to a given query q, the ANN algorithm consists of returning any point in P that lies in a certain radius from q. This technique reduces the time and resources needed to find a good enough neighbor of q [13]. Annoy, or Approximate Nearest Neighbor Oh Yeah, is a ready-to-use implementation of ANN developed and made available by Spotify [1].

Lastly, GPT (Generative Pretrained Transformer) has gained prominence for its ability to generate coherent and contextually appropriate text based on a given prompt. In the context of text recommendation, GPT can be used not only to suggest similar content but also to generate summaries, explanations, or new content that complements the learner's current material. This generative capability allows for a highly personalized and adaptive learning experience, where the content evolves in response to the learner's interactions and progress [14].

[1] https://github.com/spotify/annoy

4.1 BERT-based multi-label classification

For the first machine-learning-based solution, we opt for multi-label classification (MLC) as a straightforward way to recommend all aspects that apply to the given task. This is achieved by finetuning a pre-trained BERT-based language model on the task-aspects pairs of our training sets.

4.1.1 Training

A pre-trained BERTForSequenceClassification was finetuned on our datasets −the German and the English tasks separately. The inputs to these models are the tasks' descriptions and the labels are the aspect IDs of the training set. Using the sigmoid function as activation function, the model outputs 0 or 1 for every label. It is judicious to note that we had to reduce the threshold of the sigmoid function to 0.2 because higher thresholds yielded no results.

The German model was finetuned using 3942 tasks and vectors of 896 labels. The English model was finetuned using 6270 tasks and vectors of 882 labels. Table 4.1 shows the training evaluation of the two models.

	train loss	**accuracy**
German	0.044	0.99
English	0.045	0.98

Table 4.1: Training evaluation of the BERT MLC models on German and English data.

4.1.2 Testing and evaluation

Before running the models on the test set, we had to discard from the latter the aspects that are not in the training set. This step is necessary but it affects the quality of the results.

Both models act like dummy models: they always output the same vector. This behavior is reflected in the evaluation metrics stated in Table 4.2. The precision of 0.3 in both models indicates that when a label is predicted as positive, it is only correct in 30% of the cases. The recall being equal to ≈0.4 means that only 40% of the positive labels are retrieved. The two values directly result in low F1-scores. The null accuracy means that none of the test label vectors was predicted correctly which is expected with such long vectors.

	Precision	**Recall**	**F1-score**	**Accuracy**
German	0.30	0.36	0.33	0.0
English	0.30	0.44	0.36	0.01

Table 4.2: Evaluation of the BERT MLC models on German and English data.

The length of the label vectors coupled with a small number of data entries makes the multi-label classification severely challenging. On top of that, these vectors are highly sparse: the maximum number of aspects per task is 80, i.e. less than 10% of the labels are positive, and 60% of the tasks are attributed only 1 aspects resulting in vectors of more than 800 0's. These are two major factors to explain the poor results of the suggested solution.

4.2 Nearest neighbor on BERT embeddings

This method works more like a text search-engine: for a given task, we look for the most similar task from the train set and recommend its aspects. To do this, we trained BERT-tiny on triplets of the format (*anchor*, *positive*, *negative*) using the triplet-loss function. Then Annoy, the Approximate Nearest Neighbor index from Spotify, was used to fetch the task that is most similar to the input.

4.2.1 Methodology

Training BERT-tiny. Training BERT-tiny with triplet-loss function using as inputs (*anchor*, *positive*, *negative*) makes the word-embedding vector of the reference text (*anchor*) closer to the embedding vector of the matching text (*positive*) and further away from that of the non-matching or dissimilar text (*negative*). This technique aims at producing word-embeddings that are more accurate in the situation at hand because two tasks can be considered similar for having similar sets of aspects despite the difference in their texts, and vice-versa.

The triplet-loss function minimizes the distance from *anchor* to *positive* and maximizes the distance between *anchor* and *negative*. To generate the triplets from the train set, we iteratively sample a random *anchor*, select a *positive* as a task that has at least 90% of common aspects with the *anchor* but not 100% (*anchor* and *positive* do not have the same set of aspects), and select a task that has 0 common aspect with the *anchor* to be the *negative*. This computation outputs tens of millions of triplets, we sample 1'000'000 for each language for the training. Table 4.3 states the evaluation metrics of the models training as output by TripletEvaluator. The perfect accuracy means that upon training all the entries have a smaller distance between *anchor* and *positive* than between *anchor* and *negative*.

	train loss	**accuracy**
German	0.085	1.0
English	0.065	1.0

Table 4.3: Training evaluation of the BERT-tiny models on German and English triplets.

After training, the models are used to compute the word embeddings of the tasks in the training set later used in the building of the Annoy index.

Annoy. For the recommendation of aspects we rely on nearest neighbor algorithm. Annoy allows to build the index once and save it on disk for future searches which makes the computations faster and less costly. The index is a forest of 10 trees containing *(id, embedding_ vector)* for each training task, where *id* is later used for cross-referencing to retrieve the task's aspects.

4.2.2 Testing and evaluation

For a given task, we compute its word-embedding using the pre-trained language model, then we retrieve the task with the most similar embedding in the Annoy index. After that, we fetch this recommended task's aspects and suggest them as aspects to the input. For evaluation, we compute the Jaccard similarity between the suggested and the true aspects of the given task. Table 4.4 shows the results of this evaluation.

	min	**max**	**mean**	**median**	**% 0**	**% 1**
German	0	1	0.57	0.67	25%	22%
English	0	1	0.52	0.80	40%	32%

Table 4.4: Statistical values of the Jaccard similarity between the true and the suggested aspects. % 0 and % 1 columns represent the percentage of the test entries with similarity equal to 0 and to 1 respectively.

We observe a significant improvement in the average values of the Jaccard similarity compared to the ones obtained with LDA and LSA. Also, the fact that half of the entries have a score higher than 0.67 in German and 0.8 in English is very promising. In addition to that, more than 20% of the entries are attributed a set of aspects that is identical to their true set in both languages. The number of the recommended sets that are completely disjoint of the true sets is still large, but given the rest of the results these occurrences might need a closer evaluation of their quality since a lot of similar aspects have different IDs.

4.3 Generative AI: GPT-3.5

All the aforedescribed solutions recommended aspects only via their IDs. Coming to the last solution, we rely on the generative AI to recommend aspect descriptions for the input tasks.

4.3.1 Methodology

We use GPT-3.5-turbo through calls to the OpenAI API to generate descriptions of aspects. With some prompt engineering, we succeed to have suggestions of descriptions of aspects that are inspired from given train examples. We sample 70 to 100 pairs of (task description, list of aspects' descriptions) with each call, and ask GPT to suggest aspects for a given test task. The used prompt is:

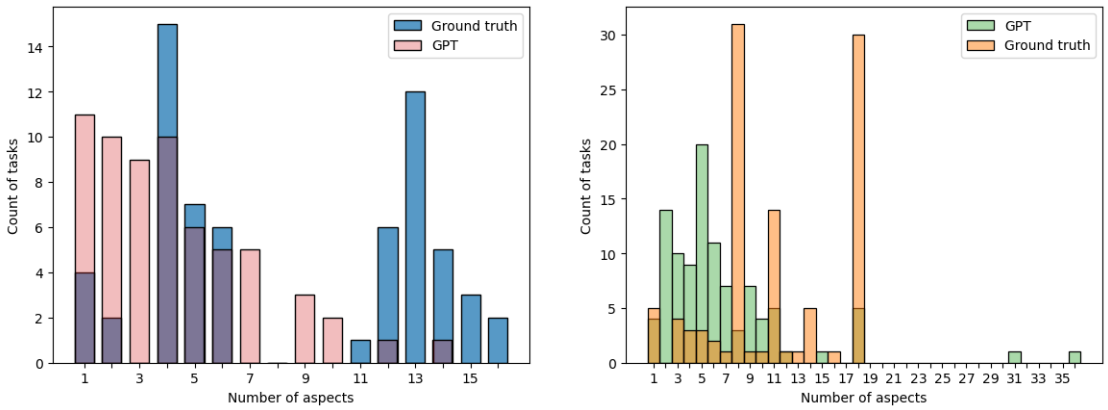
```
You get inspired from the given examples to suggest aspects for the
following task: {test_text}. Give your answer in the format
'[aspect1, aspect2, ...]'.
```

The list of examples is given in the format:

```
Text:{task} [ASP] Aspects:{aspects}.
```

4.3.2 Testing and evaluation

In most cases, GPT suggests fewer aspects than the ground truth as visualized in Figure 4.1. In fact, GPT suggests 4 aspects in German and 6 in English on average, i.e. up to 50% less than the average number of true aspects.



(a) German data (b) English data

Figure 4.1: Histograms of the number of aspects per task suggested by GPT compared to ground truth in both languages.

Evaluating this model's performance is challenging because there is no straightforward way to compare the generated text to the aspects' descriptions. To have a glimpse of GPT's output we present a few examples.

This first example shows good suggestions:

Task	Translate "Tom isst eine Banane." to English.
True aspects	['The ability to convert text from one language to another.', 'The ability to use the right input format.', 'The ability to correctly match the verb to the subject.', 'The ability to answer in the right language.', 'Can correctly spell written text.', 'The ability to convert text from one language to another.', 'The ability to correctly match the verb to the subject.', 'Can correctly spell written text.']
GPT aspects	['The ability to convert text from one language to another.', ' The ability to use the right input format.', ' The ability to answer in the right language.', ' Can correctly spell written text.', ' The ability to correctly match the verb to the subject.']

GPT's suggestions can either be very concise compared to the true aspects:

Task	Schreibe was du hörst auf Englisch.[2]
True aspects	['The ability to convert text from one language to another.', 'The ability to write the correct amount of requested information.', 'The ability to correctly match the verb to the subject.', 'The ability to use the right input format.', 'The ability to answer in the right language.', 'Can correctly spell written text.', 'The ability to formulate the answer in the correct tense.', 'This aspect applies when the students writes nonsense.', 'This aspect is here to help the student.', 'Detector: Theory', 'The student knows the word stay.'...]
GPT aspects	['Language proficiency', 'Listening comprehension', ' Spelling accuracy', 'Content understanding', 'Verb-subject agreement']

or more specific like in this example:

Task	Translate the following sentence to English: "Wir sitzen vor dem Computer".
True aspects	['The ability to convert text from one language to another.']
GPT aspects	['The ability to convert text from one language to another.', 'Using the right input format.', 'Correctly matching the verb to the subject.', 'Answering in the right language.', 'Ability to spell written text.']

Sometimes GPT cannot contain the urge of answering the given question instead of suggesting aspects:

[2]Write what you hear in English

Task	Nenne das Formelzeichen und die Einheit des elektrischen Widerstands.[3]
True aspects	['Die/Der Lernende gibt keine leere Antwort ab.', 'Die/Der Lernende verwendet angemessene Sprache.'][4]
GPT aspects	['R', ' Ohm']

Another challenge has to be pinpointed, that of the post-processing of GPT's answers. Despite being specifically asked to give the answer in the form of a list, GPT gets creative sometimes and changes this pattern which makes it difficult to capture the answers. Here is an example of a creative answer that is post-processed poorly: `['Based on the provided examples', ' here are potential aspects for the task: 1. Describing the Weekend: - Being able to communicate in the past tense. - Providing details on the locations visited...]`.

4.4 Conclusion

Finetuning BERT for multi-label classification did not yield reliable results. This is due to the large number of labels compared to the small size of the dataset, the sparsity of the label vectors and to the imbalance of the labels distribution in the dataset. MLC in principle could resolve this text recommendation exercise, but it needs to be coupled with other techniques either to further pre-process the dataset before training or to reduce the labels. Unsupervised clustering of aspects' word-embeddings as shown in Appendix B could be implemented to group the labels and reduce the sparsity.

The text-search-based solution generated the best results among all the solutions. With median score exceeding 0.6 in both languages and attaining 1 for more than 20% of the test data, this method can be part of a more elaborate solution.

The GPT-based solution have interestingly intelligent outcomes and is capable of handling new types of tasks unlike the other solutions but it comes with the challenges of the prompt engineering and the post-processing of the responses. With the right tweaking, generative AI models could provide high quality results but in the long run this would increase the number of different aspects exponentially making it harder to manage. Plus, this generated data cannot be included in future model training because of the curse of self-consuming AI [15, 16, 17].

[3] Give the symbol and unit of electrical resistance.

[4] ['The learner does not give a blank answer.', 'The learner uses appropriate language.']

Chapter 5

Discussion and Future Work

In the presented work, LDA and LSA as topic modelling techniques did not perform well on the short texts of tasks. The detected topics were overlapping and in many cases did not capture the essential purpose of the task (translation, hearing, comprehension, writing...). As stated by Yan et al. in [18], the fundamental reason why conventional topic models do not work well on short texts is that they implicitly capture the document-level word co-occurrence patterns to reveal topics and thus suffer from the data sparsity of short documents. In future work, one can explore the biterm topic model (BTM) proposed in [18] as a novel way to model topics in short texts. BTM has also been the subject of later literature aimed at making it faster [19], and improving it with word embeddings in various applications[20, 21].

The solution based on multi-label classification (MLC) did not yield acceptable results because of the very high number of labels compared to the small size of the training data. This method could be improved by additional pre-processing steps to reduce the number of labels. Methods based on label correlation, label ranking or subset sampling were proven effective to improve the performance of MLC models [22, 23]. Another solution could be to couple the MLC model with a label classifier to determine the number of labels to output as suggested by Azarbonyad et al. in [24].

Finetuning BERT-tiny on triplets of tasks and using the Annoy index as a search engine to recommend aspects is what worked best in this setup, given the shortness of the texts and the size of the dataset. The similarity between the predicted and the true sets of aspects marked high scores compared to LDA- and LSA-based solutions. This score being computed on sets of IDs, the cases of zero similarity cannot be prevented. With the reduction of similar aspects to one unique aspect, the cases where the true and the predicted sets are completely disjoint should decrease drastically.

Using generative AI to recommend aspects tailored to the given task is an interesting solution. It is intelligent and could adapt to new types of tasks. But the costs of prompt engineering and the post-processing of the responses in addition to

the curse of self-consuming models prevent this solution from being the ideal one. GPT's future could be exploited as a subpart of a bigger solution to make it faster or improve the output's quality.

Lastly, the presented work and the future improvements to be implemented can be used as a building block to solve the problem of aspects recommendation for tasks of type CLOZE.

Appendix A

LSA Labels Mapping

Table A.1 shows the probability of each task topic to be assigned any aspect topic and the threshold t. Highlighted are the aspect topics with probability $p > t$, thus mapped to the task topic on the same row.

taskTopic	aspectTopic	aspectCount	totalCount	probability	$t = 1/n$
0	**0**	492	637	**0.772**	1/2
	1	145		0.227	
1	**0**	279	369	**0.756**	1/2
	1	90		0.243	
2	**0**	116	147	**0.789**	1/2
	1	31		0.210	
3	**0**	304	441	**0.689**	1/2
	1	137		0.310	

(a) German data

taskTopic	aspectTopic	aspectCount	totalCount	probability	$t = 1/n$
0	**0**	397	533	**0.744**	1/5
	1	34		0.063	
	2	50		0.093	
	3	15		0.028	
	4	37		0.069	
1	**0**	85	170	**0.500**	1/5
	1	7		0.041	
	2	38		**0.223**	
	3	20		0.117	
	4	20		0.117	
2	**0**	373	542	**0.688**	1/5
	1	27		0.049	
	2	61		0.112	
	3	25		0.046	
	4	56		0.103	
3	**0**	375	523	**0.717**	1/5
	1	33		0.063	
	2	54		0.103	
	3	15		0.028	
	4	46		0.087	
4	**0**	63	86	**0.732**	1/5
	1	2		0.023	
	2	11		0.127	
	3	4		0.046	
	4	6		0.069	

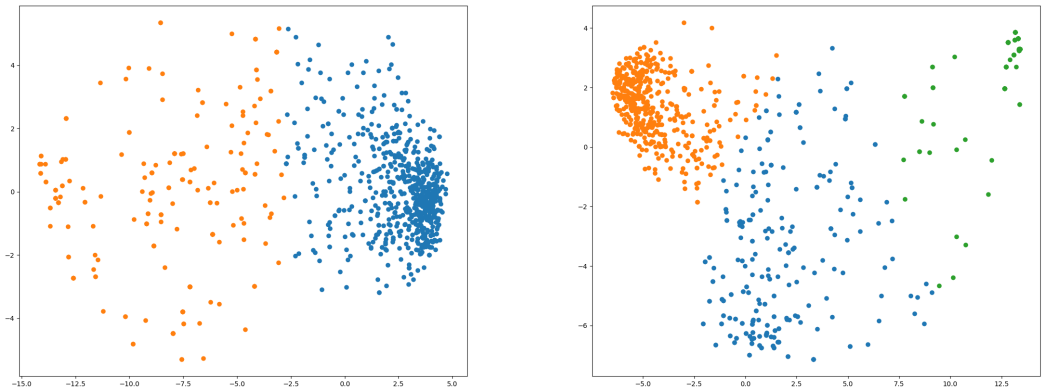
(b) English data

Table A.1: Computation details of the second method of LSA labels mapping.

Appendix B

Clustering of aspects' descriptions

These clusterings explore the diversity of the aspects' descriptions and can be used to reduce the number of labels for multi-label classification. The scatterplots present the result of using K-Means on the word embeddings reduced to two dimensions using principal component analysis (PCA). The word embeddings were comupted using BERT-tiny trained on triplets (*anchor, positive, negative*) like in Section 4.2. 2 clusters are detected in German aspects and 3 in the English ones.



(a) German data

(b) English data

Figure B.1: Results of clustering aspects' word-embeddings.

Bibliography

[1] J. W. Gikandi, D. Morrow, and N. E. Davis, "Online formative assessment in higher education: A review of the literature," *Computers & education*, vol. 57, no. 4, pp. 2333–2351, 2011.

[2] R. Kundal, "Review of trending systems for automatic assessment and scoring of student answers," *GLS KALP: Journal of Multidisciplinary Studies*, vol. 4, no. 2, pp. 16–24, 2024.

[3] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (K. Inui, J. Jiang, V. Ng, and X. Wan, eds.), (Hong Kong, China), pp. 6382–6388, Association for Computational Linguistics, Nov. 2019.

[4] R. Pramana, Debora, J. J. Subroto, A. A. S. Gunawan, and Anderies, "Systematic literature review of stemming and lemmatization performance for sentence similarity," in *2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA)*, pp. 1–6, 2022.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[6] Q. Chen, L. Yao, and J. Yang, "Short text classification based on lda topic model," in *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 749–753, 2016.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[8] N. E. Evangelopoulos, "Latent semantic analysis," *WIREs Cognitive Science*, vol. 4, no. 6, pp. 683–692, 2013.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019.

[10] I. Chalkidis, E. Fergadiotis, P. Malakasiotis, and I. Androutsopoulos, "Large-scale multi-label text classification on eu legislation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6314–6322, 2019.

[11] P. G. Syriopoulos, A. D. Andriopoulos, and D. A. Koutsomitropoulos, "Evaluation of language models for multilabel classification of biomedical texts," in *Artificial Intelligence Applications and Innovations* (I. Maglogiannis, L. Iliadis, J. Macintyre, M. Avlonitis, and A. Papaleonidas, eds.), (Cham), pp. 68–78, Springer Nature Switzerland, 2024.

[12] I. Turc, M.-W. Chang, K. Lee, and K. Toutanova, "Well-read students learn better: On the importance of pre-training compact models," 2019.

[13] A. ANDONI, P. INDYK, and I. RAZENSHTEYN, *APPROXIMATE NEAREST NEIGHBOR SEARCH IN HIGH DIMENSIONS*, pp. 3287–3318.

[14] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, (Red Hook, NY, USA), Curran Associates Inc., 2020.

[15] S. Alemohammad, J. Casco-Rodriguez, L. Luzi, A. I. Humayun, H. Babaei, D. LeJeune, A. Siahkoohi, and R. G. Baraniuk, "Self-consuming generative models go mad," *arXiv preprint arXiv:2307.01850*, 2023.

[16] M. Briesch, D. Sobania, and F. Rothlauf, "Large language models suffer from their own output: An analysis of the self-consuming training loop," *arXiv preprint arXiv:2311.16822*, 2023.

[17] I. Shumailov, Z. Shumaylov, Y. Zhao, Y. Gal, N. Papernot, and R. Anderson, "The curse of recursion: Training on generated data makes models forget," 2024.

[18] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 1445–1456, 2013.

[19] X. He, H. Xu, J. Li, L. He, and L. Yu, "Fastbtm: Reducing the sampling time for biterm topic model," *Knowledge-Based Systems*, vol. 132, pp. 11–20, 2017.

[20] X. Li, A. Zhang, C. Li, L. Guo, W. Wang, and J. Ouyang, "Relational biterm topic model: Short-text topic modeling using word embeddings," *The Computer Journal*, vol. 62, no. 3, pp. 359–372, 2019.

[21] J. Huang, M. Peng, P. Li, Z. Hu, and C. Xu, "Improving biterm topic model with word embeddings," *World Wide Web*, vol. 23, no. 6, pp. 3099–3124, 2020.

[22] Q. Gu, Z. Li, and J. Han, "Correlated multi-label feature selection," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 1087–1096, 2011.

[23] X. Che, D. Chen, and J. Mi, "Label correlation in multi-label classification using local attribute reductions with fuzzy rough sets," *Fuzzy Sets and Systems*, vol. 426, pp. 121–144, 2022.

[24] H. Azarbonyad and M. Marx, "How many labels? determining the number of labels in multi-label text classification," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 10th International Conference of the CLEF Association, CLEF 2019, Lugano, Switzerland, September 9–12, 2019, Proceedings 10*, pp. 156–163, Springer, 2019.