Research Master Thesis

Matching Ontologies in the Education Domain with Semantic Similarity

Adrielli T. Lopes Rego

a thesis submitted in partial fulfilment of the requirements for the degree of

MA Linguistics
(Human Language Technology)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities

Supervised by: Lisa Beinborn (VU), Thijs Westerveld (Wizenoze)
2^{nd} reader: Ilia Markov (VU)

Submitted: June 29, 2022

Abstract

In this digital age of information overload, aligning education curricula can facilitate the sharing of education resources and consequently the curation of digital information according to teaching and learning needs. Cross-curriculum alignment is challenging, as learning objectives are often defined and structured differently across curricula, depending on the education model or national education system, for instance. This thesis addresses the task of matching learning objectives across education curricula. I propose a model based on semantic similarity, with the focus on finding representations that can capture the semantic relations relevant for matching. In addition, the information contained in lower and higher layers of the curricula are used to enrich representations. The results indicate that using fine-tuned contextualized embeddings result in better matching than baselines, with further improvements when information from lower layers is added to the learning objectives of the source curricula. No improvement is seen when adding information from higher layers. Matching learning objectives seem to require knowledge that goes beyond semantic similarity. Structure-based methods and other relevant information than the curriculum node texts may be needed to account for domain- and application-specific matching.

Declaration of Authorship

I, Adrielli Tina Lopes Rego, declare that this thesis, titled *Matching Ontologies in the Education Domain with Semantic Similarity* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 29 June 2022

Signed:

Acknowledgments

I would like to thank my supervisors Dr. Lisa Beinborn and Dr. Thijs Westerveld, who supported this work with insightful feedback and inspirational discussions. I am deeply grateful to Carsten Schnober and Gerben de Vries from the Wizenoze Science team, for the numerous feedback and helpful suggestions. I also thank Somya Agrawal, a lead expert curator at Wizenoze, who contributed with essential insights and annotations. I am indebted to my supervising mates Marcel, Charlotte and Elisa for peer-reviewing, and my second reader Ilia Markov for taking time to critically analyse this work. I am forever grateful to both my families, the Brazilian one and the Dutch one, for all your love and support. And finally, I am specially grateful to my partner Yaman and our dog Malú. I could have never done this without your daily doses of emotional support. Gratitude is the word of the day.

List of Figures

Contents

Chapter 1

Introduction

Learning and teaching have been through a major digital revolution in the 21st century. With the advances in knowledge sharing through the internet, the Web has become the number one place to access information, including educational content, in most countries. This digital age allows for access to an unprecedented amount of information, raising the need for bringing structure in light of the information overload. In the context of education, this translates into organizing the information such that digital content can more easily and trust-worthily support teachers and students in their educational journey.

One solution proposed in Educational Technology (EdTech) is to closely align educational content on the web to education curricula. An education curriculum describes learning objectives, that is, knowledge and skills expected to be met by a target learner. However, learning objectives may be defined and structured differently across education curricula depending on various factors, such as the national education system and specific education model. This curriculum heterogeneity hinders information sharing and re-use across curricula, creating a need for computational models that can automatically match the curricula. The resulting alignment can facilitate sharing education content across curricula and consequently the curation of information according to teaching and learning needs.

The task of aligning education curricula can be seen as *ontology alignment*. This study investigates the task of matching education curricula, i.e. ontologies in the education domain, and proposes a model based on *semantic similarity* for this task. The following section briefly introduces the task.

1.1 Semantic Similarity for Ontology Alignment

Ontologies can be defined as "a set of assertions that are meant to model a particular domain of discourse" (Euzenat et al., 2007). They are typically expressed as graphs or trees with nodes as entities and links connecting entities with some relation(s). Ontologies are widely applied to model a domain of knowledge, such as the biomedical domain (e.g. Kibbe et al. (2015)), with concepts and relations between those concepts, facilitating information retrieval.

In addition, ontologies can support knowledge sharing and re-use. They frequently differ, however, in how they model a particular domain. For instance, ontologies modeling the same domain may use different terms for similar concepts, use the same terms to refer to dissimilar concepts, and classify similar concepts differently (Euzenat et al.,

2007). This heterogeneity raises a challenge for knowledge sharing and re-use between ontologies. Ontology alignment addresses this challenge. This is the task of generating correspondences between ontologies to allow for "knowledge and data expressed in the matched ontologies to interoperate" (Euzenat et al., 2007), favoring the joint consideration of resources between ontologies (Ardjani et al., 2015).

The nodes of an ontology are typically extremely short texts, used as labels for concepts or classes. Matching nodes from different ontologies minimally requires comparing their textual strings. An intuitive approach to matching text is to measure the similarity between them. Matching can thus be formulated as some similarity computation between the nodes of the ontologies under comparison (Ardjani et al., 2015). Because of the lack of context, accurately representing the meaning of such texts to compute similarity can be challenging (Han et al., 2021). In addition, the short texts are usually in a specific language variety or domain according to the ontology application(s). Matching such texts may require capturing in-domain[1] meaning (dis)similarity, as well as user requirements particular to the use case. In the next section, the use case on which this thesis is based is briefly described.

1.2 Use Case: Matching Learning Objectives

In the education domain, a curriculum with learning objectives may be represented as an ontology. Representing a curriculum as an ontology allows for various educational applications, such as aligning educational content in the web to learning objectives specified in a curriculum. In such application, when a new curriculum is added, it may be desirable to automatically match the learning objectives of the new curriculum to the learning objectives of the existent curricula. This matching supports content curation by allowing the re-use of educational content. This is the use case of this study, which comprises the Wizenoze curriculum collection.

The Wizenoze curricula are structured in a pre-defined tree format, in a taxonomy fashion. The nodes in the same depth of the tree constitute a layer. The following layers form the curriculum tree, from highest to lowest: `CURRICULUM`, `GRADE`, `SUBJECT`, `UNIT`, `TOPIC` and `QUERY`. Each curriculum is defined as an ordered sequence of such layers. The tree is filled according to the original curriculum, thus different curricula may be have different tree depths and widths. A part of the curriculum tree with *Cambridge* as original curriculum is illustrated in Figure 1.1. In this example, `GRADE` *IGCSE* (International General Certificate of Secondary Education) contains *Biology* and *Physics* as `SUBJECT`. In *Biology*, *Characteristics and classification of living organisms* is a `UNIT`, of which *Characteristics of living organisms* is a `TOPIC`. This `TOPIC` contains the curated `QUERY` *Respiration*. Importantly, nodes in the `QUERY` layer correspond to the learning objectives of the original curriculum manually optimized to retrieve relevant educational resources for the users.

A major challenge for cross-curriculum matching is that the curricula are highly heterogeneous, with diverse and subjective choices as to what concepts to cover and how to label and structure them. In other words, they may contain semantic, terminological and structural differences relative to each other. Therefore, matching the curricula minimally requires identifying linguistic relations between concepts of the *tar-*

[1] In this study, "domain" is loosely used to refer to data containing terms related to specialized knowledge. However, there is no precise definition of "domain" in NLP, and this discussion is outside the scope of this study.

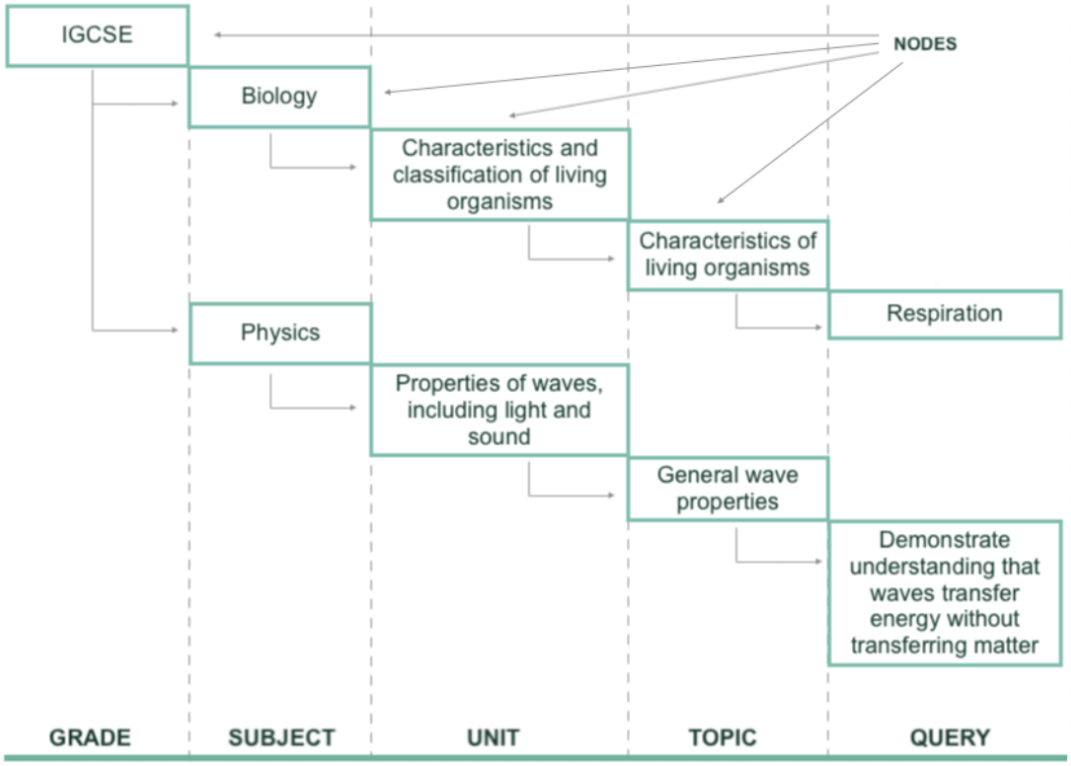


Figure 1.1: Example of curriculum tree, extracted from the Wizenoze Curriculum API page. *IGCSE* (International General Certificate of Secondary Education) is a `GRADE`, of which *Biology* and *Physics* are `SUBJECTS`. *Characteristics and classification of living organisms* is a `UNIT` in *Biology*, and *Characteristics of living organisms* is a `TOPIC` in this `UNIT`. This topic contains the curated `QUERY` *Respiration*.

get curriculum, i.e. the new curriculum to be added to the collection, and the *source curricula*, i.e. the existent curricula in the collection. Here the focus is directed to semantic relations between concepts, using semantic similarity computations to match learning objectives across curricula.

Examples 1 to 4 are concepts, or learning objectives, from different education curricula. Different methods to compute similarity between them can yield different matches. Measuring similarity with string overlap would result in 3 and 4 having high similarity to 1, as each contains the term *pressure*. But in fact, 1 refers to a different, although related, sense of *pressure* than those in 3 and 4. Similarity based on semantics, rather than the string, is required to avoid such incorrect matches.

1. *Root Pressure* (Class 11 > Biology > Plant Physiology > Transport in Plants)
2. *root pressure and guttation* (Class 11 > Biology > Plant Physiology > Transport in Plants)
3. *Blood Pressure* (Class 6 > Biology > Human Body > Circulatory System)
4. *Pressure* (Class 8 > Physics > Force and Pressure > Pressure)

To overcome the limitations of string overlap, matching is treated as a pairwise semantic similarity computation. That is, each learning objective in the target curricu-

lum, i.e. the *anchor* text, is paired with each learning objective in the source curricula, i.e. the *candidate* texts. The candidates with the k highest similarity scores to the anchor are considered a match. In such set up, a main challenge is to find representations that can capture the meaning of learning objectives such that relations that define matches and mismatches can be derived.

A relatively recent method to generate representations from text is to use transformer-based language models pre-trained on a huge amount of data (e.g. Devlin et al. (2018)). They are not only able to surpass string forms, but also to create representations that are *contextualized*. Pre-trained contextualized representations have the advantaged of incorporating the textual context of each word occurrence. In addition, they can be fine-tuned to the domain and task at hand. Thus they can, in principle, better capture the meaning of learning objectives as they occur in the linguistic context, relative to non-contextualized representations.

Information from the other layers of the curriculum may also aid matching. 1 and 2 are a match and they belong to similar `TOPIC`, `SUBJECT` and `GRADE` nodes (Transport in Plants, Biology and Class 11). In contrast, the non-matches 3 and 4 belong to dissimilar `TOPIC` and `GRADE` nodes (Human Body, Class 6) and dissimilar `TOPIC`, `SUBJECT` and `GRADE` nodes (Force and Pressure, Physics, Class 8), respectively. Another main challenge is thus to combine information from other layers of the curriculum with the representations of the learning objectives to support matching.

The higher layers of the curricula may support matching by providing coarse-grained information on the learning objectives. For instance, the grade can suggest the difficulty level of the educational content targeted by the learning objective. This information can be useful for matching, as a pair of learning objectives from similar grades are more likely to match than a pair from different grades. In contrast, the lower layer of the source curricula, that is the webpages with educational content, may describe or explain terms in the learning objective from the source curricula, addressing the lack of context in the source curricula. Thus the ontology in which the learning objectives are embedded may help yield useful matches.

1.3 Research Questions

This study is part of the growing field on how Natural Language Processing techniques can be used for aligning ontologies. I focus on semantic similarity between learning objectives to determine a match, and apply different techniques for combining information from other layers of the curricula.

The aim of this study is twofold. First, the encoders that generate meaningful representations for the learning objective texts need to determined. "Meaningful" here means that the representations can capture semantic relations and user requirements that potentially define a match. Second, given the lack of word context in the learning objectives, using the information in the other layers of the curricula may be essential to improve matching. As mentioned before, fine-grained information on the terms of the learning objective may be derived from lower layers. Since it is only available for the source curricula, it may enrich the representations of the learning objectives in the source curricula, addressing the lack of context in the source side. In contrast, coarse-grained information may be stored in the higher layers of both target and source curricula and it may help capturing domain- and application-specific (dis)similarities. These aims entail the following research questions:

(**RQ1**) *Do contextualized representations improve matching of learning objectives across education curricula, compared to non-contextualized representations?*

(**RQ2**) *Does adding fine-grained information from the lower layers of the education curricula improve matching?*

(**RQ3**) *Does adding coarse-grained information from the higher layers of the education curricula improve matching?*

To my knowledge, no previous study has explored matching learning objectives across multiple education curricula. Moreover, this seem to be the first study to approach ontology matching in the education domain with semantic similarity using contextualized embeddings.

1.4 Outline

The remainder of this thesis is structured as follows. Chapter 2 provides a background on semantic similarity of short texts and introduces previous studies on using ontology information to improve matching. Chapter 3 describes the data used, including the curriculum structure and cross-curriculum heterogeneity observed. Chapter 4 presents the task formulation, match definition, an overview of the proposed system and models used, and the experimental set up. Chapter 5 reports the results of each main experiment targeting each research question. Chapter 6 provides a detailed analysis of the results, including possible effects of in-domain factors on performance and an annotation study of the types of hits and errors of each model. Chapter 7 discuss the methods and results, pointing out the main issues and directions for future work. Finally, Chapter 8 contains the main conclusions of this thesis.

Chapter 2

Literature Review

Finding the most similar learning objectives from source curricula given a learning objective of a target curriculum can be seen as a *one-to-many* alignment problem. Alignment can consist of different components. First, text needs to be converted into a representation that allows for executing a semantic similarity operation. Second, semantic similarity needs to be computed between text representations. The fundamental problem of measuring semantic similarity between texts is presented in section 2.1. Different approaches for text representation are described in section 2.1.1 and for computing semantic similarity in section 2.1.2.

Short texts may lack sufficient context for accurate matching. Other nodes in the hierarchical structure in which texts are embedded may be informative for alignment. Different methods to incorporate this information are discussed in section 2.2. Techniques for enriching representations are presented in section 2.2.2. Given that the learning objectives are embedded in education curricula, matching may be approached as an ontology alignment between education curricula. Ontology alignment studies are discussed in 2.2.3.

2.1 Semantic Similarity of Short Texts

Learning objectives in education curricula are short texts. Matching learning objectives across curricula minimally requires comparing their textual strings. An intuitive approach to matching text is to measure the similarity between them. Measuring textual similarity is one of the most fundamental tasks in NLP and forms the basis to various downstream tasks. Similarity can be computed at various levels of granularity, e.g. word, sentence, short texts, paragraph, and document level. One way to define semantic similarity at a more abstract level is as the degree of meaning equivalence between two linguistic expressions (Agirre et al., 2015). What precisely constitutes meaning remains challenging to define.

In linguistics, defining linguistic meaning and similarity has been subject of much debate between theoretical frameworks. A consensus among linguists is meaning depends on various linguistic levels. Not only the surface forms, that is, the sequence of sounds, in the case of speech, or characters, in the case of text, but also, and foremost, larger units of language are indicative of meaning (dis)similarity. These units can be morpheme(s), word(s), sentence(s) and discourse, which combined form the meaning of a linguistic expression. This feature shared by all human languages is known as the principle of *compositionality* (Baker and Hengeveld, 2012).

In NLP, much effort has been put into capturing the meaning of linguistic expressions while being input to a computational model. This usually involves converting text or speech to numbers which computers can use. Measuring similarity in this context means comparing the resulting representations. How well the representations capture relevant meaning that goes beyond word level can be a determiner of system performance. Short text spans can be particularly challenging because they usually lack linguistic context that can be informative for determining meaning (Han et al., 2021).

At least two factors are important for measuring semantic similarity: how the text is represented and which function to use for computing semantic similarity. Before computing semantic similarity between texts, they need to be converted into a representation which can be input to semantic similarity computations.

2.1.1 Techniques for Text Representation

Because text is unstructured, noisy data, it needs to be pre-processed before it can serve as input to a computational model. Substantial research effort has been directed to converting text into representations that can be input to computations, yet semantically meaningful. How well these representations can capture the meaning of the corresponding text is vital for success in measuring semantic similarity.

Perhaps the most straightforward way to represent text is using their surface forms, as strings of characters. In early work on measuring textual similarity, text was mainly represented as a string of characters, and string-overlap metrics were used to determine similarity, such as Dice's coefficient (Dice, 1945).

An obvious limitation of using strings to represent text is that it does not capture meaning (dis)similarities beyond what is determined by the surface form of linguistic expressions, relying on exact matching to retrieve texts. For instance, the learning objectives in 5 and 6 do not have similar meanings, even though both contain the word "matter". In contrast, the learning objective in 7 can be considered somewhat similar to that in 5, as they are in the same topic, but do not have any word overlap. A string overlap metric would erroneously match 5 to 6 and not to 7.

5. Dual Nature of Radiation and Matter
6. Why They Matter - The Sensory Systems
7. De Broglie's Wavelength

To overcome the limitations of treating text as a sequence of n-gram characters, vectors emerged as a more "meaningful" input representation. Vectors are perhaps the most common format to represent text in computational models. In linear algebra, a vector is defined as a line with length and direction. Vectors can be manipulated with mathematical operations such as concatenation and averaging, and can be compared using distance or closeness measures. The intuition is to use the word's context to produce a vector such that words that occur in similar contexts have similar vectors (Jurafsky and Martin, 2021). Units beyond the word-level can be represented by combining the vectors of the composing words with averaging or summation, for instance.

The conversion from text to vector(s) can take various forms. A major distinction in the literature is that between one-hot and dense vectors. One-hot vectors are sparse, as each feature, e.g. words, corresponds to a binary dimension in the vector. In dense vectors, on the other hand, each feature is embedded in a d-dimensional space and

corresponds to a vector with d entries in this space (Goldberg, 2016). Another more recent distinction is that between static and contextualized embeddings. Both are dense, low-dimensional vectors, but while static embeddings map each word *form* to a vector irrespective of its different senses, contextualized embeddings map each word *occurrence* to a vector, such that the word itself and its context are considered.

Early work encoded text using one-hot encoding, but such sparse vectors turned out to not be very meaningful, mainly because they cannot capture dependencies between linguistic features, such as words (Goldberg, 2016). In contrast, dense, embedded vectors represent words such that words with similar contexts are close in the embedding space. To compute such representations, a model needs to be trained to learn to map each word to a dense vector. One of the first techniques generated static word embeddings with unsupervised approaches based on language models. These techniques learn values for each word embedding by either predicting the word given its context or the context given the word (e.g. Mikolov et al. (2013)).

One problem with static embeddings is that they may not capture the aspects of similarity required to perform the target task with the target data (Goldberg, 2016). Additionally, each word form in the training data is mapped into a single embedding for all contexts in which it occurs (Torregrossa et al., 2021). This means that they are not sensitive to context. Polysemous words have their multiple meanings reduced into one representation. Another issue is that static embeddings fail to capture the meaning of groups of words and wrongly assume that each context word always equally contributes to the meaning of the target word (Torregrossa et al., 2021).

Contextualized Embeddings

To address these issues, contextualized embeddings were proposed (e.g. Peters et al. (2018), Devlin et al. (2018)). They are dynamic word representations that vary according to the specific input sequence in which they occur. One of the first successful contextualized embeddings was ELMo (Embeddings for Language Models) Peters et al. (2018). Each word representation considers the left-to-right and right-to-left contexts. The word representations are the weighted sum of the two internal layers of a bidirectional LSTM. A disadvantage of ELMo is that it is not deeply bidirectional, as the combination of left and right contexts occurs as a post-processing concatenation.

BERT (Bidirectional Encoder Representations from Transformers), was proposed to overcome this limitation (Devlin et al., 2018). BERT pre-trains representations from unlabeled text using auto-encoders aimed at reconstructing the original data from corrupted input sequences. This allows for conditioning the representations to both left-to-right and right-to-left contexts simultaneously (Torregrossa et al., 2021). BERT uses stacked encoders from the transformer architecture by Vaswani et al. (2017). The model contains a multi-head self-attention mechanism which can essentially capture the importance of each context word relative to the word being currently processed. BERT representations can be used in downstream tasks as either input features or through fine-tuning the model parameters to the target task.

Beyond Word Level

With word embeddings, measuring similarity is relatively straightforward, by for instance computing the cosine similarity between the word embeddings. For input units longer than one word, various techniques can be used to aggregate the embeddings of the

individual words. Wieting et al. (2015) compared six different solutions for representing sentences and found that using an LSTM to generate sentence embeddings performed well in in-domain settings for semantic similarity, while mean pooling outperformed the LSTM on out-of-domain settings. Previously to transformers, the state-of-the-art for learning sentence embeddings was InferSent (Conneau et al., 2017). It employs labelled Natural Language Inference (NLI) datasets to train a siamese BiLSTM, topped by a max pooling operation. The resulting vectors are concatenated with their element-wise difference and product and fed to a classifier.

Following the success of transformer-based pre-trained language models in capturing word meaning for downstream tasks, proposals have been put forward on how to use or modify such models to encode sentences or short texts. A simple and widely used approach is to input the text into a transformer-based model, and then average the output layer, or use the output of the first token, i.e. CLS token. However, this technique has been shown to result in inferior embeddings for measuring semantic similarity with cosine, sometimes even underperforming averaged static embeddings (Reimers and Gurevych, 2019).

To address this limitation, various proposals of transformer-based models for encoding sentences and short texts emerged. One of such proposals is Sentence-BERT (SBERT) (Reimers and Gurevych, 2019). SBERT was proposed as an optimized BERT model for sentence embedding generation. The model employs annotated NLI datasets to train a siamese or a triplet BERT network structure. During training, SBERT consists of two to three BERT-like base models that share weights, topped with a pooling operation, i.e. CLS token, average, or max of output vectors. The resulting encodings are concatenated with their element-wise difference and input into either a softmax, cosine similarity or euclidean distance operations. In a series of experiments on Semantic Textual Similarity (STS) tasks and transfer learning tasks, the authors showed that SBERT model outperformed other state-of-the-art sentence embedding methods.

2.1.2 Approaches to Measuring Semantic Similarity

Once text is encoded, the encodings are input to a semantic similarity computation. In order to compute semantic similarity between learning objectives, the approach for measuring semantic similarity must be defined. Approaches to measuring semantic similarity can be divided into supervised and unsupervised, depending on whether annotated data is needed. Additionally, when encoding the texts to be compared, they can be encoded separately with bi-encoders, or jointly with cross-encoders.

At a higher-level, determining the best matches for a given learning objective may be seen as a search-like task. Given a learning objective as anchor text, the task is to find the most similar learning objectives in a set of of candidate learning objectives. Dense retrieval combines semantic similarity with search using bi-encoders to encode text pairs and cosine similarity to compute semantic similarity.

In the next sections, I contrast unsupervised and supervised methods for semantic similarity modeling, as well as the cross-encoding and bi-encoding methods for encoding text pairs. Lastly, I zoom in to the framework of dense retrieval, on which the method of this thesis is based.

Unsupervised Semantic Similarity

Unsupervised approaches for measuring semantic similarity between texts either rely on external resources or on vector similarity functions. As such, unsupervised techniques are commonly classified into knowledge-based and corpus-based (e.g. Han et al. (2021)).

Knowledge-based metrics rely on the structure of a Knowledge Base to compute semantic similarity. Knowledge Bases are sources containing structured knowledge representation in the form of terms or entities connected through (semantic) relations. Semantic similarity can be measured by quantifying the path length of concepts in the Knowledge Base, the shared information between them, and/or their common attributes (Chandrasekaran and Mago, 2021). A downside of such metrics is that they ignore word order and word context (Han et al., 2021), which can be relevant for measuring semantic similarity of short texts.

Corpus-based metrics in turn use the information from corpora to measure semantic similarity. Cosine similarity is the most widely used metrics to measure semantic similarity between texts (Chandrasekaran and Mago, 2021). Given two texts represented as vectors, this metric computes the cosine of the angle between the vectors projected in a multi-dimensional space. The cosine of the angle is the normalized dot product of a pair of vectors. The smaller the angle between the two vectors, the closer they are in a shared space. Two vectors are considered similar when the cosine similarity is close to 1, and considered dissimilar when close to -1.

In a unsupervised set up using a corpus-based metric, similarity estimation can be formulated as follows: $P(similar = 1|d_i, q) = \phi(\eta_q(q), \eta_d(d_1))$, where ϕ is the cosine similarity function, q is the anchor text d_1 is a candidate text in a corpus, and η is the encoder or transformation function that maps q and d to vector representations (Lin et al., 2021).

Supervised Semantic Similarity

Another approach to computing similarity between texts is to use a machine learning model which, given texts as input, outputs a score that can be interpreted as the similarity between them. Such model needs to be supervised, that is, trained with labeled data for similarity. Each input instance is labeled as to whether they are similar, in the case of classification, or as to how similar they are, in the case of regression. In a classification set up, similarity can be estimated as follows: $P(similar = 1|d_1, q)$, where d_1 is a candidate text in a corpus and q is the anchor text (Lin et al., 2021).

The distinction between using cosine function or a classifier to compute similarity is usually related to the distinction between bi-encoders and cross-encoders. In semantic similarity approaches, bi-encoders are typically used in a cosine similarity set up, while cross-encoders are typically used in a classifier set up (Lin et al., 2021). In the context of semantic similarity, contextualized embeddings for each text pair may be generated jointly, with cross-encoders, or separately, with bi-encoders.

Contextualized Embeddings in Pairwise Semantic Similarity

Typical cross-encoder and bi-encoder approaches are illustrated in Figure 2.1. In a typical cross-encoder set up, the model classifies whether a text matches a given anchor text. This means that each anchor text and candidate text are paired and jointly passed to a transformer-based model with a classification head on top, whose output is

interpreted as a similarity score. The texts are input as follows: $[CSL]q[SEP]d_i$, where $[CLS]$ is the classifier token and $[SEP]$ is the separator token. With bi-encoders, the model generates dense representations of anchor and candidate texts separately, and similarity can be computed directly via cosine similarity.

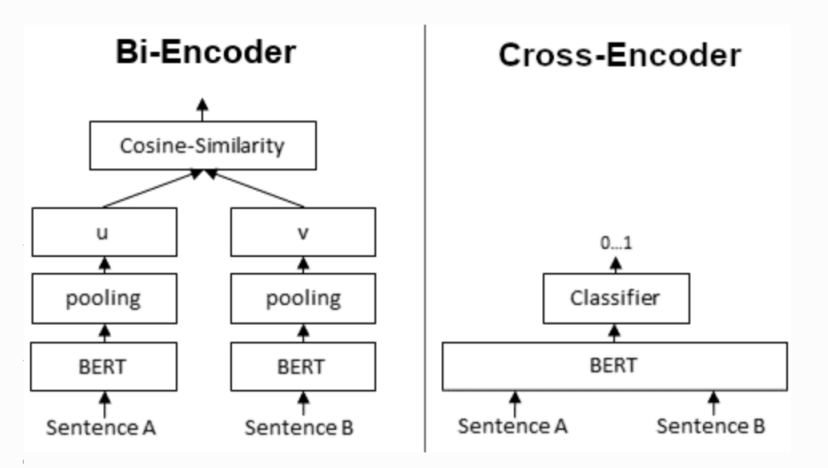


Figure 2.1: Bi-encoder vs. Cross-encoder architectures, extracted from SBERT documentation webpage.

Because bi-encoders allow anchor and candidate texts to be computed separately, bi-encoders result in faster semantic similarity computations. Even though cross-encoders have been shown to have high accuracy in measuring semantic similarity between texts (Nogueira and Cho, 2019), they are significantly slower to compute. This is because pre-encoding the texts is not possible, as both anchor and candidate texts are inputted together in a cross-encoder. Also, using the cosine function on the cross-encoder vectors has been shown to be insufficient for accurate similarity computation, performing worse than averaged GloVe embeddings (Reimers and Gurevych, 2019).

Dense Retrieval

The use of transformer-based bi-encoders with cosine similarity for search is referred as Dense Retrieval. Dense retrieval techniques measure semantic similarity directly on the vector representations generated by a transformer-based model. Contrary to keyword search, dense retrieval techniques are able to address the vocabulary mismatching problem between anchor and candidate texts, as relevance scores are computed directly from a shared vector space designed to capture semantic meaning.

As Lin et al. (2021) explain, the basic setup of dense retrieval is to learn a transformation on anchor and candidate texts. The model must convert the sequence of tokens into fixed-sized dense vectors, such that the similarity between the vectors is maximized when the anchor and candidate text match and minimized when they do not match, as measured by a similarity metric. At inference time, the goal is to retrieve the top k texts from the corpus with the highest similarity to the anchor text, according to the similarity function and using the trained encoders. If annotated data in the target domain is available, the transformer model can be fine-tuned to the target domain on similar and dissimilar texts. The outputs of the fine-tuned model are learned representations for the purpose of matching.

With this set up, the candidate text representations are independent from the anchor text representation and can be pre-computed and stored, making a mostly expensive encoding process a pre-processing stage and thus accelerating search at inference time. Another advantage is that a similarity function like cosine similarity is fast and computing it over a large collection of texts can be optimized via neighbourhood search methods (Lin et al., 2021).

The simplest transformers for ranking with dense learned representations are bi-encoders. This diverges from the typical cross-encoder transformer architecture in text classification. As explained before, in cross-encoders, the two texts are concatenated into an input template from which the model directly outputs a similarity or relevance score. In contrast, bi-encoders take the two texts as input separately and compute a representation for each, which can then be compared via a similarity function such as cosine. Since similarity highly relies on the vector representations in dense retrieval, texts must be encoded such that accurate similarity or relevance scores can be derived directly from a vector closeness metric.

SBERT is an example of such bi-encoder model designed to generate meaningful representations of sentences for large-scale text similarity comparisons. SBERT address the problem of high computational cost shown in fine-tuned BERT for pair classification by computing vectors at sentence level, as opposed to token-level, that can be compared via cosine similarity with reasonable accuracy. Although there have been various proposals after SBERT (e.g. BertFlow Li et al. (2020) and TSDAE Wang et al. (2021)), I opt for using this model for its design simplicity, implementation easiness, spread use, and focus on sentence similarity rather than retrieval problems.

The first research sub-question of this thesis addresses the effectiveness of using transformer-based models for matching learning objectives across education curricula. Dense retrieval is a suitable approach to the task, because it combines cosine similarity and bi-encoders. Cosine similarity does not require in-domain annotated data, and bi-encoders are better suited for tasks where multiple inferences are required. Therefore, this study's methodology is based on dense retrieval.

2.2 Incorporating Ontology Information

As mentioned in section 2.1, matching short texts can be challenging because they may lack context that is informative for deciding whether they are sufficiently similar. A solution to improve matching is to incorporate the knowledge from a relevant ontology. The term *ontology* originates from philosophy, where it can be defined as a systematic account of existence (Guarino, 1998). In artificial intelligence, ontology is a systematic description of a given field, built for allowing knowledge representation and conceptualizations required to meet various challenges, such as information retrieval and integration Sayed and Al Muqrishi (2017).

Ontologies differ in the level of conceptualization and expressivity. In terms of conceptualization, Guarino and Giaretta (1995) classifies ontologies into top-level, domain and application ontologies. Top-level ontologies contain more general notions, domain ontologies depict concepts relevant to a particular domain, and application ontologies represent knowledge based on both a particular domain and a particular task.

Studies on using knowledge from ontologies typically focus on web search using top-level or domain ontologies written in RDF or OWL (Ramkumar and Poorna, 2014). However, application ontologies can also be used. The education curricula in the use

case of this study can be considered application ontologies, as they are built specifically for the domain and task at hand. Retrieving educational content in the web based on such education curricula is an example of using application ontologies to retrieve relevant information.

Another relevant classification is that by Euzenat et al. (2007), which emphasizes the expressivity of ontologies, that is, how explicit and well-defined the semantics of the ontology is. In Euzenat and Shvaiko's scale of expressivity, lists of terms, such as glossaries, have low expressivity, taxonomies have medium expressivity, and formal ontologies have high expressivity. The use case of this study represents education curricula as taxonomies, where learning objectives are organized into categories in a hierarchical manner. Each layer in the hierarchy defines the semantics of the belonging nodes and holds relations with the remaining layers.

To my knowledge, no previous proposal has been put forward for one-to-many ontology alignment that augments semantic similarity estimation with information from ontologies in the domain of education. Some studies have attempted to incorporate higher layers from education curricula to aid search of educational content (e.g. Usta et al. (2021)). Other studies enrich text representations outside the domain of education (e.g. Nogueira and Cho (2019)). Yet other studies match learning objectives through mapping one education curriculum to another education curriculum, i.e. one-to-one alignment as opposed to one-to-many alignment, without semantic similarity (e.g. Lmati et al. (2015)), or with semantic similarity but not in the domain of education (e.g. He et al. (2021)). In the remainder of this section, I describe such studies, with the focus on the education domain.

2.2.1 Grade- and subject-specific ranking models

Usta et al. (2021) investigate the use of learning to rank models for search engines in the education domain. Based on previous work on query-dependent ranking models, they build a ranking model for each grade and subject. The intuition is that the grade the user is in and the subject the query targets affects the user needs represented by the query. Thus different ranking functions are needed for different grades and subjects. As expected, the specialized ranking models were found to outperform a general ranking model and query-dependent models based on automatic query clustering.

This approach is not suitable for the use case of this thesis, because different education curricula may have different conceptual formulations of grades and subjects. The same content that is addressed in grade 5 in one curriculum may be addressed in grade 6 in another curriculum. An even more fragile correspondence can be seen with subjects. The same content can be part of *physics* in one curriculum, but under *earth and space sciences* in another curriculum. The way educational content is conceptualized and structured in layers such as subject and grade may highly vary across curricula, making a meta-curriculum necessary to apply such approach.

2.2.2 Enriching text representations

This study experiments with enriching the text representations of learning objectives with information of the respective curriculum, such as the lowest curriculum node. This level is populated with educational content generated via semantic search and attested by human curation to help students achieve the respective learning objectives. They can be seen as the lowest layer in the hierarchy, i.e. education curriculum, in which

the learning objective is embedded, and are expected to enrich the learning objective representation with more fine-grained information.

Work on enriching text representations for similarity computations is mainly from query and document expansion techniques for information retrieval. Query or document expansion refers to augmenting queries or documents with additional textual representations to aid matching. Most of this work involves term expansion, that is, augmenting query and document representations with additional weighted terms (Lin et al., 2021).

How the additional terms are generated and weights are set differ across studies. For instance, Nogueira and Cho (2019) proposed document expansion via query prediction by training a sequence-to-sequence model to generate queries given a document from the corpus. The queries are then appended to the document text. Evaluation on the MS MARCO passages Bajaj et al. (2016) indicated significant improvements compared to no document expansion.

In this study, a simple approach is taken to enriching learning objectives from the source curricula. They are expanded with text from educational content resulted from curated web search. The learning objectives from the source curricula have segments from the respective educational content added to their encodings. Moreover, no weights are set for the added educational content text.

2.2.3 Ontology Alignment

The most popular methods for ontology alignment still rely on a combination of structural and logic-based matching with string-based matching (e.g. LogMap, Jiménez-Ruiz and Cuenca Grau (2011) and AML, Faria et al. (2013)). To address the problem of vocabulary mismatch between ontologies, studies may rely on external resources. Lmati et al. (2015) is an example of an ontology alignment study in the education domain that employs WordNet in the attempt to capture the meaning of strings. They align two domain ontologies in the form of education curricula by combining three techniques: string-based, WordNet-based and structural.

In string-based matching, the entity strings are "standardized" with a series of NLP pre-processing techniques, such as lowercasing. To partially overcome the limitations of string-based matching, they measured the similarity between the sets of corresponding WordNet synsets. Finally, to incorporate structure, graph-based metrics are used to measure how similar the two entities are positioned in their respective ontologies, with the assumption that the entities are similar if their neighbourhoods are similar. The authors compared the resulting mappings with manually created mappings and found reasonable precision and recall scores for the nodes of some layers, such as Module and Chapter.

The first attempt to introduce word embeddings to ontology alignment was that by Zhang et al. (2014). The proposed model uses pre-trained word embeddings to represent the entities under comparison and then computes similarity between the vectors with Euclidean distance. Structure and domain-specific semantics are ignored, and each pairwise match is independent of each other. Follow-ups to this work addressed these gaps by, for instance, adapting word embeddings to the target domain and task and incorporating structure via an extension of the stable marriage algorithm (Kolyvakis et al., 2018).

With the advances brought by transformers-based models in text representation, attempts began to emerge to exploit BERT-like models for ontology alignment. To my

knowledge, the first successful proposal is BERTMap (He et al., 2021). BERTMap is a multi-stage system that combines BERT embeddings for mapping prediction with structural and logical information for mapping refinement. Fine-tuning BERT is preceded by a corpus construction stage for sampling of positive and negative examples. Positive examples are pairs of entities that are annotated in the ontology or in external resources as being synonyms, whereas negative examples are pairs annotated as non-synonyms. Given sets of synonyms and non-synonyms, BERT is fine-tuned to classify whether a pair of entities in the ontologies are synonyms.

Inference time comprises a mapping prediction stage and a mapping refinement stage. During mapping prediction, candidate entities are selected for a given target entity via ranking on idf-based scores of sub-word inverted indices. These candidates are then scored with string-matching. If string matching does not return any matches, the fine-tuned BERT is used. The resulting mappings are then input to mapping refinement.

Mapping refinement contains two sub processes, mapping extension and mapping repair. The mappings are extended with an iterative mapping extension algorithm that applies the same mapping prediction method to the parents and child nodes of the entities under comparison. Only the extension mappings which achieve a 0.9 score threshold are kept. Addionally, a mapping repair stage removes the mappings that lead to logically inconsistencies when integrating the two ontologies.

In incorporating information of the higher layers in the education curricula to match learning objectives, I experiment with a re-ranking method inspired on the mapping expansion module proposed in BERTMap, explained in section 4.4.3.

Chapter 3

Data

The data used in this thesis is part of the *Wizenoze* collection. This collection currently contains 28 education curricula in English stored in a standard, pre-defined tree format. Education curriculum describes the knowledge and skills expected to be met by a target learner. The curricula stored in the Wizenoze database vary in the content covered and their structure. For the purposes of this study, a sub-set of k-12 curricula was selected. Following the definition of the US education system, k-12 curricula are curricula that target learners from kindergarten to 12th grade (ages 4 to 18). They are assumed to cover more similar concepts relative to each other than non k-12 curricula.

I describe the data in more detail in the following sections. Section 3.1 presents descriptive statistics of the sub-set of k-12 curricula, with a focus on cross-curriculum variation in structure and content. Section 3.2 details the train, development and test datasets.

3.1 Curriculum Description

In the Wizenoze collection, a curriculum corresponds to textual nodes organized in a hierarchical manner. The nodes in the same depth constitute a layer. As mentioned in section 1.2, a curriculum in the collection is composed of the layers `CURRICULUM`, `GRADE`, `SUBJECT`, `UNIT`, `TOPIC` and `QUERY`, from highest to lowest. The layers constitute a pre-defined tree format, which is filled according to the original curriculum, resulting in cross-curriculum variation in tree width and depth.

Table 3.1 provides an overview of the structure of the sub-set of k-12 curricula selected for this study. The curricula vary in width, specially at the `QUERY` and `GRADE` layers, as more variation can be seen in the number of queries and grades across curricula. All curricula have at least the `CURRICULUM` node, similar to a tree root, and the `QUERY` layer, similar to tree leaves. Some topics may not be further specified into queries, meaning that the topic name is used as a query to retrieve educational resources. Content-wise, all curricula target k-12 students and subjects mainly from the Formal Sciences (e.g. Mathematics and Computer Science) and Natural Sciences (e.g. Physics and Biology).

The curricula used in this study are listed below, with ages and subjects targeted per curriculum.

- *Indian Certificate of Secondary Education* (ICSE): for ages 7 to 17 with the subjects English, Chemistry, Physics, Mathematics, Environmental Sciences, Biology,

	ICSE	**CBSE**	**Cambridge**	**NGSS**	**CCSS**	**CSTA**	**English**	**Scotland**	**Lebanon**	**all**
Queries	5182	4490	1705	446	429	132	974	160	965	14483
Queries per topic	4	4	5	4	5	3	2	6	4	4
Topics per unit	3	3	3	3	2	3	5	4	2	3
Units per subject	7	10	5	4	2	5	5	3	4	6
Subjects per grade	4	3	4	3	5	1	3	2	3	4
Grades	12	12	6	3	5	3	7	1	7	-

Table 3.1: Descriptive statistics of curriculum trees of set of k12 curricula, with the (mean) quantity of nodes in each layer of each curriculum.

Computer Science and Science.

- *Central Board of Secondary Education* (CBSE): for ages 7 to 17 with the subjects English, Chemistry, Physics, 21st Century Skills, Environmental Sciences, Biology, Computer Science and Science.

- *Cambridge International* (Cambridge): for ages 10 to 15 with the subjects Chemistry, Computing, Physics, Mathematics and Biology.

- *Next Generation Science Standards* (NGSS): for ages 11, 12 and 16 with the subjects Physical Sciences, Earth and Space Sciences and Life Sciences.

- *Common Core State Standards* (CCSS): for ages 11 to 16 with the subjects Statistics and Probability, Algebra, Geometry Concepts, Numbers Concepts, Algebra Concepts, Measurement Concepts, Statistics Concepts, Geometry, Numbers and Quantities, and Functions.

- *National Curriculum in England* (English): for ages 8 to 15 with the subjects Design and Technology, Chemistry, Physics, Computing, Mathematics, Biology and Science.

- *National Curriculum in Scotland* (Scotland): for age 8 with the subjects Sciences, and Numeracy and Mathematics.

The text in each node is short. When tokenized with an off-the-shelf NLTK tokenizer, grades and subjects usually contain 1 to 2 tokens, whereas topics and queries have 3.7 and 4.2 tokens on average, respectively. The shortest topic and query are each 1 token long, the longest topic is 43 tokens long, and the longest query is 23 tokens long.

As mentioned in section 1.2, queries are roughly equivalent to learning objectives in the original curriculum. In case they are not exactly the same, the query is the learning objective modified by expert curators to yield relevant educational resources for the end user in a semantic search tool. Therefore, I henceforth refer to the queries in the curriculum tree as *learning objectives*. The goal is to avoid confusion with the concept of query in search. Additionally, the term *learning objective* is closer to the semantics of the text instead of its use in a semantic search tool of this particular use case.

Curriculum Content Heterogeneity Importantly, the curricula are heterogeneous. Euzenat et al. (2007) classify ontology heterogeneity in syntactic, terminological, conceptual/semantic and pragmatic. Syntactic heterogeneity is when they are expressed in different ontology languages, e.g. OWL and F-logic. Pragmatic heterogeneity occurs when semantically equivalent entities in different ontologies can be interpreted differently depending on pragmatics. More relevant to this study are terminological and conceptual heterogeneity types. Terminological heterogeneity occurs when the same entity is described with different terms across the ontologies. 8 is an example of two nodes from different curricula with similar meanings but different terms.

8. (a) Equation of the first degree
   (b) Equation of the line through two given points, or through one point with a given gradient

Conceptual or semantic heterogeneity can be further specified into differences in coverage, granularity and perspective. Differences in coverage occur when ontologies describe different and possibly overlapping domains. Some education curricula may cover a subject or knowledge that is not covered by other curricula. Granularity differences occur when the same entity is described with different levels of detail. 9 illustrates granularity differences, as 9a is more specific than 9b.

9. (a) Square roots of a real number. Powers of a real number
   (b) Representing Real Numbers on the Number Line

Differences in perspective occur when the same entity is described from different points of view. This is illustrated in 10, where 10a refers to the concept of coordinate planes from the perspective of its use in another topic than that of 10b.

10. (a) Transformations in the plane
    (b) Bases and reference frame in the plane

Specifically to the domain of education, differences in complexity may occur when two nodes describes the same entity at different levels of difficulty. 11 is an example of differences in complexity, where 11a refers to a more complex education content than that of 11b.

11. (a) Ordering of R. Intervals
    (b) Ordering of Integers

.

In sum, the curricula used in this study are a sub-set of 7 k-12 curricula from the Wizenoze collection, mainly targeting subjects from the Formal and Natural sciences. The curricula are in a pre-defined tree structure with the layers , and the learning objectives matched in this study correspond to the query nodes of the curricula. Although they are all k-12 curricula and filled a pre-defined tree format, they present variation in structure and content. More relevant for this thesis is the variation in content, which appears in the terminology used and the concepts covered. Importantly, the learning objectives are mainly short texts of 3 to 4 tokens, and conceptual differences in coverage, perspective, granularity and complexity are observed across curricula. The data used in this thesis for training, development and testing are described in the following section.

	ICSE	**CBSE**	**Cambridge**	**English**	**NGSS**	**CCSS**	**Scotland**	**total**
Train	2286	2163	680	330	154	117	69	5799
Dev	254	241	76	37	18	14	8	648
Test	636	601	189	92	44	33	20	1615
proportion	39%	37%	11%	5%	2%	2%	1%	100%

Table 3.2: Distribution of learning objectives across curricula in train, development and test sets.

3.2 Data Description

In the use case, learning objectives are used as search queries to retrieve educational resources in the web. As a result, each learning objective is aligned to at least one educational resource. In order to annotate the data for matching, I retrieve all pairs of learning objectives from different curricula which share at least one educational resource. Importantly, the educational resources are checked by human curation to ensure relevance in relation to the learning objective. When two learning objectives return at least one identical educational resource, they are assumed to be a match for the purposes of this study.

The set of learning objectives that share at least one educational resource form the semi-annotated data for this study. Each first learning objective in the pair is considered the anchor, that is, the learning objective from the target curriculum which needs to be matched. From a search perspective, they are the queries or anchor texts for which the system finds the most relevant matches in the search space. The second learning objective in the pair is considered a correct candidate given the anchor. The data is divided into train, development and test sets, following a 70-10-20 split. Each set contains anchor learning objectives from 7 different k-12 curricula. The distribution of anchor learning objectives across curricula is show in Table 3.2.

Chapter 4

Methodology

In this chapter, the methodology applied in this study is presented. Matching learning objectives across education curricula can be formally defined in various ways. I define the task and motivate the approach in section 4.1. Similarly, what a match precisely is depends on the data and task at hand, among other aspects. I discuss the definition of matching in this study in section 4.2. Once the task has been formally and conceptually defined, a high-level overview of the system proposed in this study is given in section 4.3. Finally, I turn to the experiments carried out to answer the research questions and find the best settings for the system in 4.4.

4.1 Task Formulation

Matching learning objectives across education curricula is defined as a semantic similarity task followed by ranking, in a one-to-many curriculum alignment set up. The motivation to define it as such is threefold. First, there is high variation across curricula and their learning objectives in multiple aspects, such as structure, coverage, terminology, perspective, complexity and granularity, as explained in section 3.1. This variation represents a challenge to generating accurate mappings between curricula. Semantic similarity can, at least to some extent, capture terminological and conceptual differences.

Second, typical ontology alignment studies present a one-to-one set up, where one ontology is aligned to another ontology. The use case is to match the learning objectives of a new curriculum to the learning objectives of any existent curricula in the collection, thus a one-to-many alignment. A system that generates mappings from a new curriculum to another specific curriculum is not as useful from the user perspective.

Alternatively, one could suggest an alignment between each curriculum and a standard formal ontology for education. Possibly as a consequence of high variation across curriculum, data linking of education curricula is incipient, making the availability of such standard to which to link the curricula scarce. In addition, the curricula in this study are not in an standardized format for data linking, such as RDF, raising an extra challenge for such approach.

In sum, given the high variation across education curricula, the one-to-many mapping nature of the use case and the lack of standards that can account for all variation together with the non-standard format of the available curricula, I approach matching as a semantic similarity task. As such, matching learning objectives across curricula can be formalized as follows:

Given an anchor learning objective q and a set of candidate learning objectives $D = \{d_i\}$, generate a top-k ranking of candidate learning objectives from D that maximizes a metric.

Both q and d are short texts composed of 4 tokens on average, and they are the lowest, most specific nodes of two different education curriculum trees C_q and C_d, respectively (please see section 1.1 for a detailed explanation of the curriculum tree structure and Figure 1.1 for an example). The chain in a curriculum tree to which q or d are attached can be defined as $C_i = \{t_i, u_i, s_i, g_i\}$, where t_i is the `TOPIC` node, u_i is the `UNIT` node, s_i is the `SUBJECT` node, g_i is the `GRADE` node and cr_i is the curriculum node. These are the parent nodes from a tree perspective, or classes from a taxonomy perspective. In addition to C, d is a parent node to a set of educational resources defined as $R_d = \{r_1, ..., r_5\}$, which provide more fine-grained information on the meaning of d.

For instance, given q = *Nitrogen Fixation and Nitrogen Cycle*, t_q = *Microorganisms*, u_q = *Food*, s_q = *Science*, g_q = *Class 8* and cr_q = *CBSE*. A candidate can be d = *Nitrogen Fixing Bacteria*, with t_d = *Economic Importance of Bacteria*, u_d = *Diversity in Living Organisms*, s_d = *Biology*, g_d = *Class 9* and cr_d = *ICSE*. This candidate is further linked to R_d, with r_1's title = *The stages of the nitrogen cycle*.

How similar d is to q can be estimated by comparing their vector representations with a similarity function. As suggested in Lin et al. (2021), similarity estimation can be formalized as follows:

$$S(match = 1|d_i, q) = \phi(\eta_q(q), \eta_d(d)),$$

where ϕ is the cosine similarity function and η is the encoder or transformation function that maps q and d to vector representations. η_q and η_d can either be the same or separate models. Here, $\eta_q = \eta_d$. Since it is pairwise similarity, S is computed for each combination of q and d. The candidate learning objectives d_i are then ranked according to the resulting cosine similarity scores, with the highest scores ranked first.

The research questions of this study target η, q and d. The first research question asks whether η as a transformer-based language model improves matching compared to η as a non transformer-based language model. The second research question focuses on d by investigating whether expanding it with fine-grained information, that is text from education resources R, improves matching. And finally, the third research question consists of expanding both q and d with text from the higher layers in the curriculum tree C, which should contain more coarse-grained information on the learning objective.

4.2 Match Definition

In this study, two short textual strings are compared. As mentioned before, whether they match involves the notion of semantic equivalence, roughly understood as the answer to the question "Do these short texts mean the same thing?". Semantic equivalence in itself is a hard concept to pin down, and is usually defined beforehand in a study's annotation guidelines. The data used in this study is not manually annotated for semantic matching between learning objectives, thus precisely defining a match from the data perspective can be challenging.

In information retrieval, a match is typically associated with the notion of relevance. Relevance may be affected by the semantic similarity between two texts, but it is highly conditioned to the user needs. In NLP tasks, on the other hand, a match is typically

defined as similarity in linguistic meaning and thus extra-linguistic features such as user clicks are not taken into consideration (Lin et al., 2021). Here a match between learning objectives partially corresponds to semantic similarity, but some aspects from the education domain and the application of the use case may also affect matching.

Domain- and Application-specific Match As discussed in section 3.1, the learning objectives under comparison vary in aspects such as granularity, perspective and complexity. Being able to model such differences may thus be important to yield accurate matches.

In terms of granularity, a match may depend on the extent of the conceptual overlap between learning objectives. This relates to the idea of subsumption, and the direction is relevant. In principle, when the anchor learning objective is more general than the candidate learning objective, they are likely to be a match. This is because the mapping may be one-to-many, where multiple candidate learning objectives combined have a semantic equivalence relation to the anchor learning objective. Note that the task as formulated in this study does not explicitly incorporate dependencies between pairwise matches. Thus, in principle, the model would have to learn the dependencies implicitly in order to capture this relation. In contrast, when the anchor learning objective is more specific than the candidate learning objective, the degree of conceptual overlap determines whether they are considered a match for the user, i.e. the expert curator.

In terms of perspective, when learning objectives differ in the approach they take to a common subject, they do not match. This can be a challenging aspect to learn, as learning objectives may have a high degree of token overlap and nevertheless have different meanings. Similarly, in terms of complexity, learning objectives on the same subject but that differ in difficulty level are not a match. This is a notion highly related to the domain of education, since different difficulty levels address different learners, e.g. learners from different ages. Thus what might be considered semantically equivalent is nonetheless considered a mismatch for the purposes of education.

The data used for training and, to some extent, for testing, was automatically annotated on the assumption that two learning objectives match if they have at least one educational resource in common. This is the operational definition of a match in this study and thus not necessarily corresponds to the more conceptual definition of a match, that is, involving the notions of semantic equivalence and differences in perspective, granularity and complexity discussed.

The conceptual definition of match in this study is a result of consulting the in-domain expert user, the task formulation and observing the data from a non-expert perspective. However, the fact the learning objectives share educational resources, which are fine-grained descriptions or elaborations of the learning objectives, is considered to be a relevant indicator of semantic similarity. In addition, this method reflects matching specific to the use case, that is, learning objectives match when some or all educational content used to meet one learning objective can also be used to meet the other learning objective. Since the current approach focuses on semantic similarity, one can check whether semantic similarity can be sufficient to capture this application specific matching.

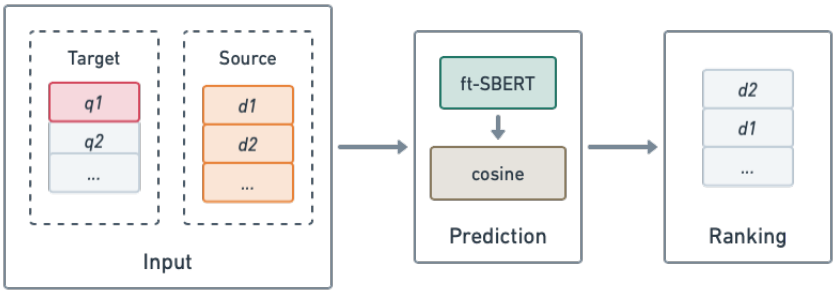


Figure 4.1: Our base system for semantic matching. Each anchor learning objective q is paired with each candidate learning objective d, separately encoded with fine-tuned SBERT and compared with cosine similarity. This is repeated for all candidate learning objectives, which are then ranked according to the resulting cosine similarities.

4.3 System Overview

The system proposed to match learning objectives in this study consists of the following three steps: pre-processing, similarity estimation and ranking. Only the first two steps are performed during training in the case of fine-tuning SBERT. All steps are performed at inference time. Figure 4.1 shows a high-level overview of the system at inference time.

In sum, the proposed system is based on SBERT encodings and cosine similarity. The task is to determine whether a candidate learning objective matches the anchor learning objective. A pair of anchor and candidate learning objectives are input to the system, which pre-process and encode them with SBERT. Then cosine similarity between encodings is calculated. For each anchor learning objective, the respective candidate learning objectives are ranked according to the cosine similarities.

4.3.1 Input Pre-processing

The text needs to be pre-processed to the format expected by SBERT before it can be inputted. Because the base model of SBERT uses BERT-like models as base model, the same input format used in BERT is required. The input sequence consists of the following format: $[CLS]$ *text* $[SEP]$, where the text is converted into WordPieces (see Wu et al. (2016) for WordPiece proposal). Longer sequences than the maximum sequence length of 128 WordPieces are truncated, and shorter sequences are padded with [PAD] token.

4.3.2 Similarity Prediction

This is the core of the system and corresponds to S and can be seen in more detail in Figure 4.2. Once pre-processed into the format required by SBERT, the input sequence is inputted to SBERT, which outputs a 384-dimensional dense representation for the whole text via mean pooling. This is computed separately for each anchor and candidate learning objectives in a pair, whose vectors are then compared using cosine similarity.

During fine-tuning, SBERT has as input the anchor learning objective, a positive example and negative example simultaneously. Positive examples are the learning ob-

jectives that share educational resources with the anchor learning objective. Negative examples are randomly samples from the remaining learning objectives. Each is inputted to a base model with shared weights in a triplet architecture. The output representations are then compared with cosine similarity, and the weights of the base models are updated according to the training objective.

4.3.3 Ranking

The resulting cosine similarities between anchor and candidate learning objectives are ranked from highest to lowest for each anchor learning objective. The top k candidate learning objectives with the highest cosine similarities with the anchor form the predicted ranking.

4.4 Experimental set up

In order to investigate the research questions of this study, experiments are conducted with η, q and d. First, the efficiency of a transformer-based η is tested against two baselines, TF-IDF vectorizer and pre-trained Fasttext encoder. This experiment is described in section 4.4.1. Second, I evaluate whether using fine-grained information from education resources manually linked to source learning objectives can improve matching. How this is carried out is explained in section 4.4.2. Lastly, I investigate different methods to use the parent nodes of the curricula to aid matching, which are described in section 4.4.3. How each setup is evaluated is explained in section 4.4.4.

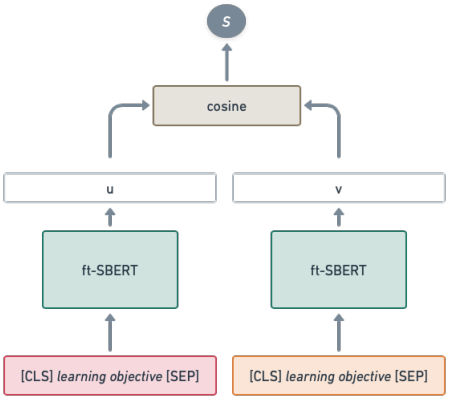
4.4.1 Semantic similarity of short texts

The first experiment round concerns changing the encoder in the prediction module of the system and evaluating how this affects the output rankings. Here the input is uniquely the text of the learning objectives, as can be seen in Figure 4.2a. As defined in section 4.1, the encoder η converts text into vector representations to be compared with a similarity function. Each model is described below.

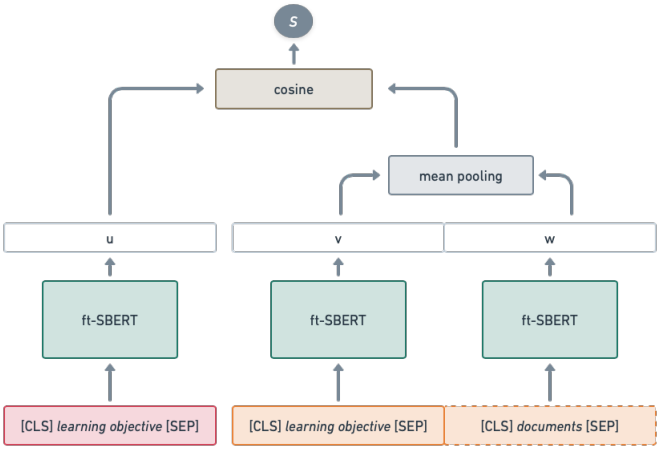
Baselines

TF-IDF stands for "term frequency - inverse document frequency" and is a widely used metric in information retrieval to encode textual input by weighting its information value relative to a document or collection of documents. The intuition is that the relevance of a term is inversely proportional to its frequency across documents. TF or "term frequency" of a word shows how often a term appears normalized by the document length. IDF or "inverse document frequency" weights the terms according to its "amount of information". Multiplying these two metrics result in the TF-IDF value. The higher the TF-IDF value, the more informative the word is to the documents it appears in (Aizawa, 2003). In this study, TF-IDF based matching is employed as a baseline. Learning objectives are encoded via TF-IDF vectorization and indexed with Elastic Search.

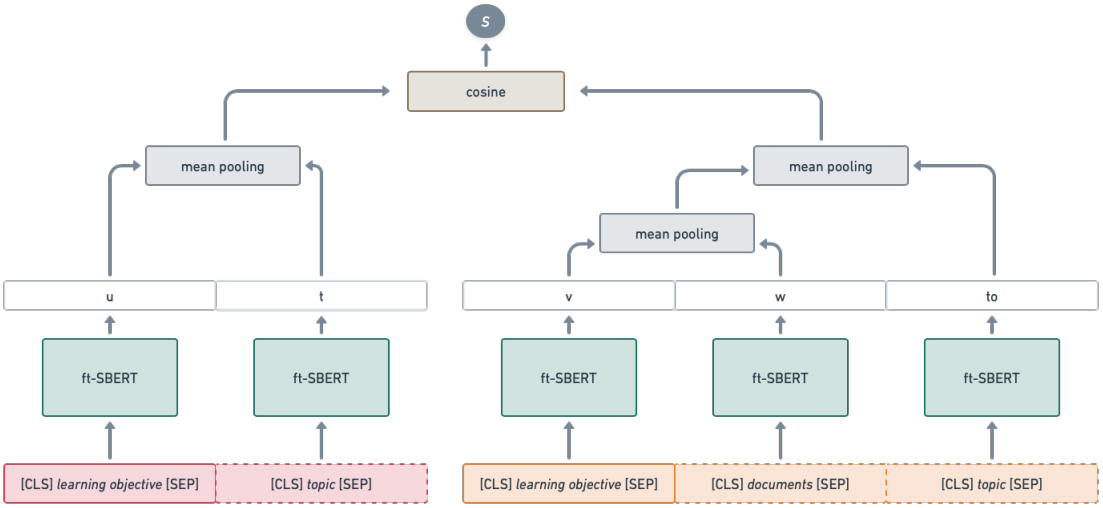
Fasttext (Bojanowski et al., 2017) is a skipgram model for representation learning where words are represented as a bag of character n-grams. The model learns a vector representation for each character n-gram, and each word is represented as the sum of



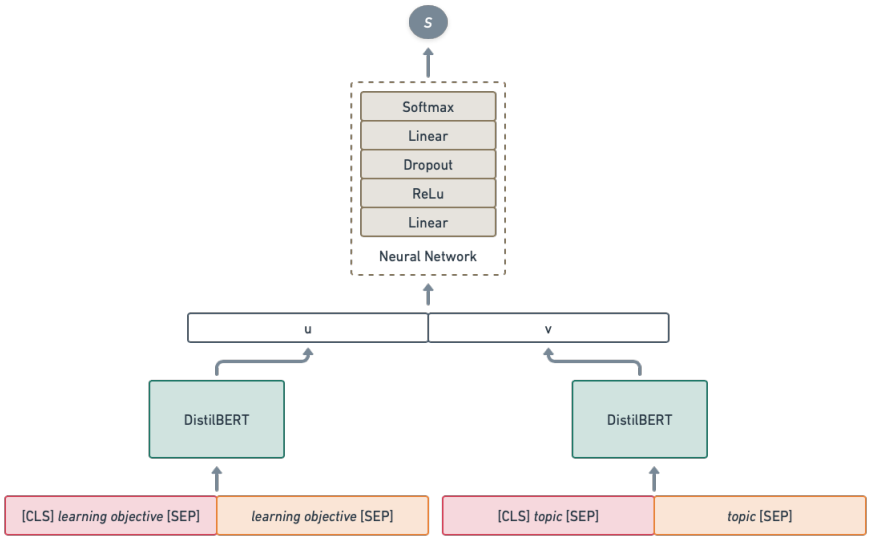
(a) Prediction module with a pair of anchor and candidate as input.



(b) Information from documents is added to candidate.



(c) Information from the respective higher layers is added to anchor and candidate. Here topic is used as an example.



(d) Bi-encoder is changed to cross-encoder and cosine is changed to neural network.

Figure 4.2: Overview of prediction module in distinct experimental settings.

such vectors. The main advantage of fasttext over other non-transformer representation learning models, such as word2vec (Mikolov et al., 2013), is that the model can generate representations for words that not appear in the training data. Pre-trained word and sub-word vectors are freely available online[1]. This study employs the 300-dimensional word embeddings with 1 million English words trained on 16B tokens from Wikipedia, UMBC WebBase corpus and statmt.org news dataset. Learning objectives are represented as the averaged vector over the vectors of the composing words.

Pre-trained SBERT

As explained in more detail in section 2.1.2, SBERT (Reimers and Gurevych, 2019) is proposed as an optimization of BERT (Devlin et al., 2018) for generating "meaningful" sentence representations that can be compared with cosine similarity. This study uses pre-trained SBERT *paraphrase-MiniLM-L6-v2*, available in the hugging face model hub[2]. This is a SBERT with a distilled version of BERT as base model and trained on paraphrase detection on out-of-domain data. It uses mean pooling to generate a fixed-size representation of the input sequence. It maps texts with a maximum sequence length of 128 tokens to 384 dimensional vectors. This pre-trained model is employed to directly encode the input, similarly to the baselines.

Fine-tuned SBERT

For further attesting the efficiency of this model for in-domain matching, fine-tuning on matching with in-domain data is performed using the automatically annotated set of learning objective pairs described in section 3.2. For that, a triplet network is used. This architecture tunes the model parameters such that distance between the anchor text and a positive example is smaller than between the anchor text and a negative example. The positive example is a learning objective that shares at least one educational resource with the anchor learning objective. Negative examples are randomly selected from the remaining candidate learning objectives. The implementation used is from sentence transformers python library with epochs = 3, learning rate = 2e-05 and batch size = 12.

To further determine the best settings for the fine-tuned model, tuning is performed on the loss function and sampling strategy. Loss functions included (a) multiple negative ranking loss and (b) triplet loss. Sampling strategies included (a) keeping target duplicates and (b) dropping target duplicates, leaving one training instance per target learning objective.

4.4.2 Incorporating fine-grained information from taxonomy

Once the efficiency of SBERT for cosine-based matching is determined compared to the baselines, text from the education resources are used to expand the candidate learning objective, as shown in Figure 4.2b. This is expected to yield better matches, given that the learning objectives are extremely short texts and may thus lack sufficient information for accurate matching. Since the educational resources are manually linked to the learning objectives by expert curation, they are assumed to provide more fine-grained information on the meaning of the learning objective. I experiment with the different

[1] https://fasttext.cc/docs/en/english-vectors.html
[2] https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L6-v2

document segments: (a) Title, (b) first sentence of summary, and (c) sentences in which at least one term of the learning objective appears. Figure shows how segments from educational resources are added to q. They are separately encoded and the resulting vectors are averaged.

Additionally, tests were conducted with different input formats and document types on the validation set. Input formats included: (a) Appended to learning objective text and encoded jointly, and (b) Encoded separately and then averaged with learning objective encoding. Document types included: (a) Pinned documents only, i.e. confirmed by expert curator as a relevant document, and (b) pinned documents plus organic documents, i.e. including documents retrieved by semantic search and not manually curated.

4.4.3 Incorporating coarse-grained information from taxonomy

To investigate whether the information of the higher layers of the curriculum can aid matching, I experiment with a few set ups. The experimentation consists of two dimensions: the method of adding the information of the higher nodes and which nodes are used.

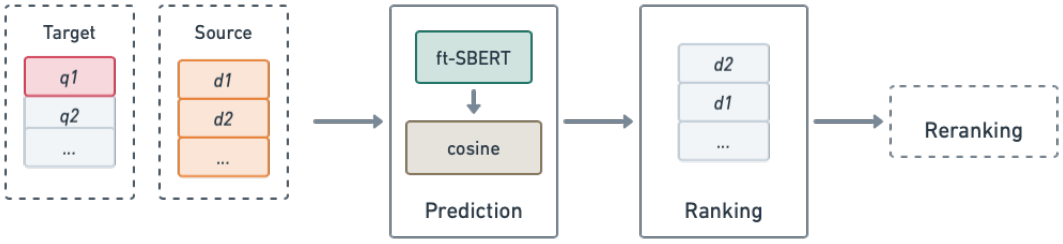
In terms of which information is added, I experiment with the topic, subject and grade layers. Grade is converted into age before being inserted into the system, because age does not have variation in names as grade does, and the mapping between grades and learner ages is readily available in the Wizenoze collection. As for the method for adding this information, three methods are investigated: enriching the input representations, re-ranking and neural classification.

Enriching Input

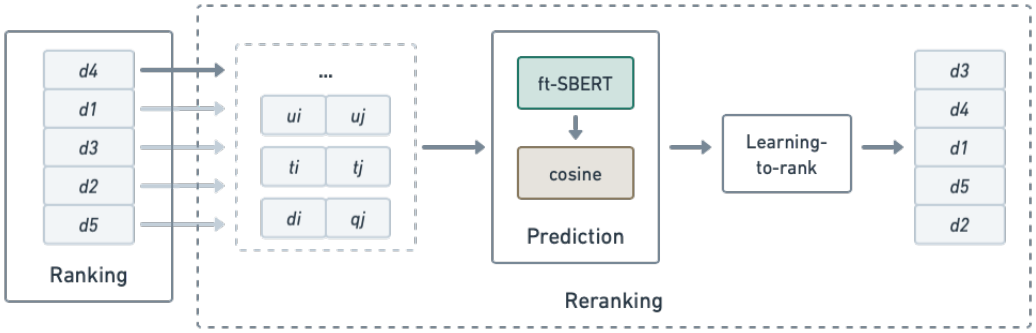
The first and perhaps simplest method is to enrich the input representations of both anchor and candidate learning objectives with the information of higher nodes. This is a similar approach as how the texts from education resources is added in that both involve enriching input representations, but they differ in how they are operationalized. Figure 4.2c illustrate this method. In adding higher layers, each node text is separately encoded. In this way, more information from each node may be captured relative to concatenating the text and then encoding all as one string. Both topic and subject are encoded with SBERT, whereas age is one-hot encoded and then converted into a dense representation with an embedding layer. The resulting vectors are concatenated, forming the representation of the respective learning objective.

Re-ranking

The second method is re-ranking based on BERTMap's mapping extension module. Similarly to BERTMap, the model applied to match the entities, i.e. learning objectives in this case, is used to compare the respective parent nodes. But instead of extending mapping as in BERTMap, the scores on the higher layers are used to refine the ranking yielded from comparing the textual representations of the learning objectives. This takes the form of a re-ranking module. This adaption is needed due to the output of the current task formulation being rankings of a specific node, i.e. learning objective, rather than graph mappings.



(a) System overview with re-ranking module.



(b) Re-ranking module in detail.

Figure 4.3: Re-ranking candidate learning objectives. t represents the `TOPIC` node and u the `UNIT` nodes. The prediction step is applied to each pair of higher nodes from anchor and candidate. The resulting cosines are aggregated with a learning-to-rank model. The candidates are then re-ordered according to the output scores from the learning-to-rank model.

Figure 4.3 illustrates the re-ranking module. Given the top k candidate learning objectives retrieved by the ranking module, re-ranking consists of retrieving the parent nodes of each anchor-candidate pair and applying S to each corresponding parent node pair. For instance, the `TOPIC` nodes of q and d are pre-processed, encoded with SBERT and compared with cosine similarity. The same can be applied for other parent nodes, such as `SUBJECT`.

When using the prediction module of the system to compare the higher nodes of the curricula, cosine similarities are returned from each pair of nodes. They then need to be combined in order to yield the refined ranking. These cosines together with the cosine from the first ranking are used as features for a learning-to-rank model, which outputs the final ranking.

The learning-to-rank model used is LambdaMART (Burges, 2010), freely available in the lightgbm library[3]. LambdaMART is a gradient boosting tree model whose goal is to estimate scores for anchor-candidate pairs such that, given an anchor, a relevant candidate and an irrelevant candidate, the relevant candidate is ranked higher than the irrelevant candidate. The model is trained with correct predictions in the first ranking as positive examples, and incorrect predictions in the first ranking as negative examples, with equal distribution between positive and negative examples.

[3]https://github.com/microsoft/LightGBM

Neural Classifier

Finally, the third method uses a neural classifier with a cross-encoder to match learning objectives. Figure 4.2d illustrates this method. Instead of encoding the texts separately and comparing the resulting vectors with cosine similarity, the texts are encoded jointly and inputted to a neural network, whose output is interpreted as a similarity score. Because automatically annotated data is available in the target domain, a classifier can be trained in a semi-supervised manner to estimate the match between paired learning objectives.

A key advantage of a neural classifier is that it is more flexible in the input and in learning different types of information for the task at hand. Each higher node can be encoded and the resulting encoding can be concatenated and inputted to a classifier head. The classifier head can then learn weights for different nodes encoded in the input vector. Additionally, information that might be better encoded with other vectorization methods than a language model, such as age, can be concatenated to the remaining node representations and added to the input of the classifier.

Because input flexibility and learning capability may be key for successfully incorporating the higher nodes of the structure, I conduct a preliminary study using a neural classifier to match learning objectives with the information of higher nodes concatenated to the classifier input. The neural network uses cross entropy loss and is composed of a linear layer, followed by a Re-lu, a drop-out and another linear layer, all from PyTorch implementation.

4.4.4 Evaluation

The desired ranking for each anchor learning objective is that matches are placed high in the ranking, within the top k. To evaluate the rankings outputted by the system, *recall@k* and *Mean Reciprocal Rank* (MRR) are used. The proportion of correct matches that appear in the top k is captured by recall@k. Recall@k is defined as the number of correct matches in the top k out of the total number of correct matches in the gold for a given anchor. For each target curriculum, recall@k is computed for each anchor and the resulting values are averaged over all anchors. For anchors with more than k matches, exceeding matches are excluded from the recall computation, i.e. the maximum denominator in the recall formula is k.

The motivation for reporting recall rather than precision is twofold. Firstly, all relevant candidates for a given anchor are known. Recall at the ranking top is not a usual evaluation metric in search tasks because it is difficult to estimate all the correct or relevant candidates given an anchor. In this study, this is possible to estimate because of the operational definition of a match for the use case, that is, two learning objectives match if they share curated educational webpages. Additionally, Precision at k is affected by the difference between the number of total number of relevant candidates for the anchor and k. The model does not have a similarity threshold and thus always returns the top k candidates with the highest similarity scores given the anchor. This means that for those anchors with fewer than k relevant candidates, precision upper bound would be lower than 1. In fact, in most cases no more than three candidates are a good match to the anchor.

Since positioning information is not captured by recall, MRR is also reported. MRR reflects the position of the first appearance of a correct match in the ranking. Reciprocal Rank is defined as $1/rank_i$ where $rank_i$ is the highest position of any correct match

for a given anchor learning objective. For instance, if a correct match appears in the first position, Reciprocal Rank will be 1, and if it appears only in the second position, it will be 1/2. If a correct match does not appear in the top k, it receives a score of zero. MRR is the averaged Reciprocal Rank over all anchor learning objectives. I report macro averages instead of micro averages across curricula, as macro averaging gives equal weight to the curricula. Since the curricula vary in size but are in principle equally important, macro averaging is a more suitable metric for this study.

Chapter 5

Results

This chapter presents the results of each experiment. All experiments were subject to three runs with different random seeds for data splitting, and model initialization in the case of training. The results shown are the mean scores over the three runs. As described in section 3.2, the test set is composed of target learning objectives from seven different k-12 curricula, automatically annotated according to the sharing of educational resources.

The outline of this chapter follows the research questions. Recall that the task is to estimate the following: $S(match = 1|d_i, q) = \phi(\eta_q(q), \eta_d(d))$. At least two components must be determined: the encoders, or η, and the information included in q and d, given that they are embedded in an informative taxonomy. Section 5.1 presents the results of different η, focusing on the effect of changing the encoder from non transformer-based to transformer-based. Section 5.2 shows the effect of enriching d with fine-grained information from education resources. Lastly, section 5.3 presents the effect of adding different parent nodes of the curriculum to q and d, as well as of various techniques to do so.

5.1 Text Encoder

The first experiment aims at answering the following research question:

(**RQ1**) *Do contextualized representations improve matching of learning objectives across education curricula, compared to non-contextualized representations?*

This experiment asks whether η as a transformer-based language model improves matching compared to η as a non transformer-based model. I experiment with pre-trained SBERT and fine-tuned SBERT vectors as contextualized representations and with TF-IDF and Fasttext vectors as non-contextualized representations.

Table 5.1 shows the results of this experiment. As expected, contextualized representations perform better than non-contextualized representations. Pre-trained SBERT significantly improves both recall@5 and MRR compared to TF-IDF and pre-trained Fasttext representations. This indicates that contextualized representations allows for better semantic matching. Fine-tuning on the target data and task, that is in-domain semantic matching, further improves both metrics. This suggests that semantic relations that may define a (mis)match are better captured with domain adaptation.

	Recall@5	MRR
TF-IDF	.34	.39
Fasttext	.36	.43
SBERT	.58	.57
Fine-tuned SBERT	**.61**	**.60**

Table 5.1: Macro averages of rankings with different text representation techniques, using uniquely the learning objective texts.

Tuning Additionally, I experimented with different loss functions and sampling strategies when fine-tuning SBERT on three different validation sets. The ones the provided the highest scores were kept for testing. Regarding the loss function, multiple negative ranking loss yielded better performance scores than triplet loss on the validation sets. Keeping the multiple negative ranking loss constant, the sampling strategy of dropping target duplicates provided higher scores than keeping target duplicates.

5.2 Incorporating Fine-grained Information

Given how short the texts are at the learning objective level, the research question is the following:

(**RQ2**) *Does adding fine-grained information from the lower layers of the education curricula improve matching?*

This question focuses on d by adding information from education resources attached to the learning objectives to d and checking whether it improves matching. Because of the small difference in performance between pre-trained SBERT and fine-tuned SBERT with only the learning objective text, I tested both models with document information.

The results with different document segments can be seen in Table 5.2. Both recall@5 and MRR are higher when any type of segment from educational resources are added. Moreover, which segment type is added does not seem to significantly affect performance. Since adding only the titles is the simplest option, this is considered the best combination of information to represent q in this experiment. This indicates that enriching the representation of the candidate texts with fine-grained information from lower layers of the taxonomy helps to capture the meaning of short texts, supporting semantic matching.

Tuning Additional experimentation was conducted with different input formats and document types on the validation set. Regarding input format, I validated on the following set ups: concatenating the texts from educational resources to the learning objective texts and encoding them jointly; and encoding text from education resources and the learning objective separately and then averaging. Separately encoding followed by averaging yielded higher recall@5 and MRR scores averaged over three different validation sets.

Regarding document types, I check whether using fewer, but of higher quality educational resources yields better matches than using more, but potentially noisy, educa-

	pre-trained SBERT		fine-tuned SBERT	
	Recall@5	MRR	Recall@5	MRR
learning objectives	.58	.57	.61	.60
+ title	.65	.62	**.67**	**.63**
+ title + sum_1sent	.65	.62	**.67**	**.63**
+ tile + sum_nsents	.64	0.62	.65	**.63**

Table 5.2: Macro averages of rankings when adding information from the lowest layer of the taxonomy, composed of educational resources. "+ title" corresponds to adding the title, "+ sum_1sent" corresponds to adding the first sentence of the summary, and "+ sum_nsents" corresponds to adding the sentences from the summary in which at least one anchor term appears.

tion resources. This question is addressed by comparing performance when using only pinned documents and when using both pinned and organic documents. As mentioned before, pinned documents are educational resources retrieved by semantic search and confirmed as relevant by an expert curator, whereas organic documents are retrieved by the same semantic search but not checked by an expert curator. After keeping the input format constant as averaged encodings, using fewer but of higher quality educational resources yielded higher performance over three different validation sets.

5.3 Incorporating Coarse-grained Information

Turning to the higher layers of the taxonomy, I ask the following:

(**RQ3**) *Does adding coarse-grained information from the higher layers of the education curricula improve matching?*

This question focuses on incorporating the nodes in the higher layers of the curricula to q and d, and evaluate whether semantic matching improves as a result. I investigate this question in two dimensions: the method with which to add this coarse-grained information, and which higher layers to add.

Table 5.3 shows the results of using different methods and adding different combinations of higher layers. The methods include enriching the representations of q and d with the texts in the higher layers (section 5.3.1), measuring semantic similarity between each higher layer and combining the cosines similarities in the form of re-reranking with a learning-to-rank (section 5.3.2), and switching the dense retrieval approach to the classical fine-tuning set up, i.e. cross-encoder plus neural classifier (section 5.3.3). The higher layers consists of topic, subject and grade. Since recall@5 and MRR seem to show the same tendencies, only recall@5 is reported in this section.

5.3.1 Enrich input with text from higher nodes

Grade is the only higher layer which does not have a negative effect on performance. All other combinations of higher layers have the effect of decreasing recall@5. Based on this experiment, I cannot conclude that the higher layers are informative.

	Enrich Input	Re-ranking	Classifier
no higher layer	.67	.67	-
+grade	**.68**	.67	-
+subject	.58	.67	-
+topic	.46	.67	-
+grade+subject	.58	.66	-
+grade+topic	.46	-	-
+subject+topc	.44	-	-
+grade+subject+topic	.46	.67	.37

Table 5.3: Macro averages of recall@5 for each method and combination of higher layers.

Another possible reason for the results is that the method used is not suitable. With this method, each higher layers has the same weight on the similarity score, when in fact it is more likely that they have different informative contributions to the similarity between anchor and candidate texts. Methods that can learn weights for the different layers may be more suitable for semantic matching between taxonomies. This idea is tested with a learning to rank model and a neural classifier in the next sections.

5.3.2 Semantic similarity of higher nodes with Re-ranking

Again, adding higher layer information does not seem to improve the results. Even though binary loss decreases on validation during training, MRR remains stable, suggesting lack of learning for the desired task. In all combinations, the cosine between queries received the highest feature importance, as measured by the number of tree splits, followed by age, topic and subject. Almost no difference in ranking metrics is seen in despite that. The results of this experiment do not support the idea that the higher layers of the taxonomy aid semantic matching between learning objectives across taxonomies.

5.3.3 Cross-encoder with Neural Classifier

Because of the computational overhead of the neural classifier — it takes 1.5 hours on average to compute similarity for each anchor learning objective —, only preliminary testing is carried out with a subset of the test set. A total of 50 learning objectives were randomly selected from the curricula, with the distribution across curricula proportional to the size of each curriculum. Due to the long training time, the neural network is only initialized once, data splitting is only performed once, and only one combination of higher layers is investigated.

Using age, topic and subject in combination with the learning objective text and document titles, the classifier achieved .37 on mean recall@5. In contrast, binary accuracy on the validation set is the impressive score of .98. Again, a discrepancy is observed between binary metric measured on validation set during training and the ranking performance at inference time. During training, the model has to learn which

candidate learning objective is a match and which is not, given an anchor. The candidates are likely to be highly different from each other, as the wrong candidate, or negative example, is randomly sampled. During testing, on the other hand, the models needs to capture much more subtle differences between candidates to retrieve the correct matches within the top k.

5.4 Summary

Contextualized embeddings were found to outperform the baselines for semantic matching, with further improvement with in-domain and task adaptation through fine-tuning the embedding model SBERT. In addition, enriching the candidate representations with fine-grained information from a lower layer, i.e. curated educational documents, was found to further improve matching. Conversely, no method nor combination of higher layers as features improved results, indicating that information from the higher layers of the curriculum does not aid matching with semantic similarity.

Chapter 6

Error Analysis

In order to gain further insight into model performance and capabilities in relation to the research questions, a qualitative analysis of the results is conducted. The analysis consists of three parts. First, I compare each model's performance in relation to in-domain factors, which include subject, age and curriculum. This is intended to verify whether the results observed in the previous chapter hold for all subjects, ages and curricula in the data. Second, correct matches and incorrect matches are randomly sampled and categorized into types, and the proportion of the each type for each model is calculated. This analysis aims at shedding light on the hits and errors of each model. Combined with a manual inspection of the samples, as well as as of a random sample of predicted rankings, strengths and weaknesses of the best performing model are suggested.

6.1 In-domain Factors on Performance

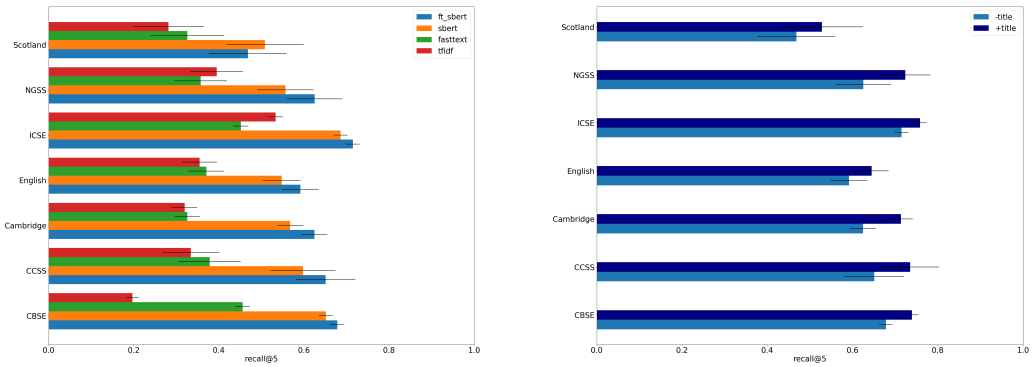
To verify whether performance across models differ depending on certain in-domain properties of the learning objectives, mean recall@5 is computed per property value across models. The properties correspond to parent nodes of the learning objectives, more specifically the curriculum, age and subject. This analysis is performed on the results of each model when using only the learning objective text, as well as when fine-grained information is added to the candidate learning objective representations.

The results on the different encoders can be seen in Figures 6.1, 6.2 and **??**. Fine-tuned SBERT produces the highest scores overall. There are no specific subjects, age ranges or curricula on which the baselines outperform, supporting the superiority of contextualized embeddings for semantic matching of short texts in the education domain. In the following sections, a more detailed analysis is provided for each in-domain factor.

6.1.1 Curriculum

Regarding curriculum, Figure 6.1 shows that all models, except TF-IDF, provide the highest scores on matching learning objectives from the ICSE and CBSE curricula, whereas Scotland is the most difficult curriculum to be matched. Higher scores for ICSE and CBSE may be due to the fact that the ICSE curriculum was created by copying part of the learning objectives of the CBSE curriculum, facilitating matching

between these curricula. TF-IDF yields the highest variation in score across curricula, indicating less robustness in matching when the curriculum changes.



(a) Recall of each model for each curriculum.

(b) Recall of fine-tuned SBERT with and without document titles for each curriculum.

Figure 6.1: Recall@5 for each curriculum.

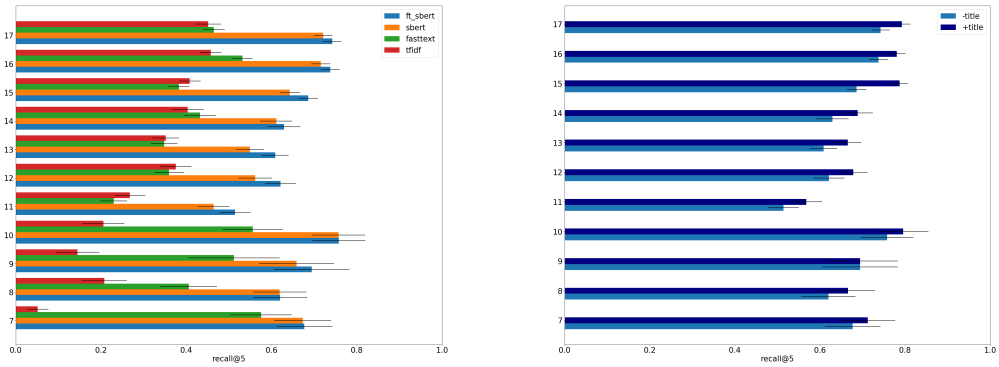
When adding fine-grained information, Figure 6.1b shows increased performance for all curricula. The lowest increase in performance is observed for the CBSE and ICSE curricula. As already mentioned, these curricula are partially copies of each other, and thus uniquely using the learning objective texts seem to suffice more often for those curricula. For the other curricula, there is a higher boost in performance, resulting in less variation in performance across curricula when learning objectives are expanded with fine-grained information.

6.1.2 Age

Regarding age, one could imagine that the higher the age at which the learning objective aims, the more complex it may be and the more specialised knowledge might be required. This would mean that learning objectives targeting older ages could be more difficult to correctly match. This is not observed in the matching performance across ages. As Figure 6.2a shows, performance does not decrease as a function of age increase in any model.

In fact, the opposite pattern is observed, with increasing performance as age increases. This may be due to the degree of standardization of the curricula content. As age increases, the learning objectives and educational content tend to be more standardized. Therefore, less cross-curriculum heterogeneity is observed, which may explain the increase in match scores as age increases. An exception to this trend is between age 10 to age 11, which could be explained by the fact that the large majority of learning objectives from ages 7 to 10 belong to the CBSE and ICSE curricula, which yield higher scores, as explained above. From age 11 onwards, other curricula which seem harder to match are included, such as NGSS and Cambridge, causing a decrease in performance score.

Performance increases for all ages when fine-grained information is added. Figure 6.2b reveals that the pattern observed with using only the learning objectives, that is the higher the age, the higher the performance, is less pronounced here. This suggests



(a) Recall of each model for each age.

(b) Recall of fine-tuned SBERT with and without document titles for each age.

Figure 6.2: Recall@5 for each age.

that the effect of standardization on matching is weakened when the learning objective terms are expanded with more fine-grained information, providing a more robust model across ages.

6.1.3 Subject

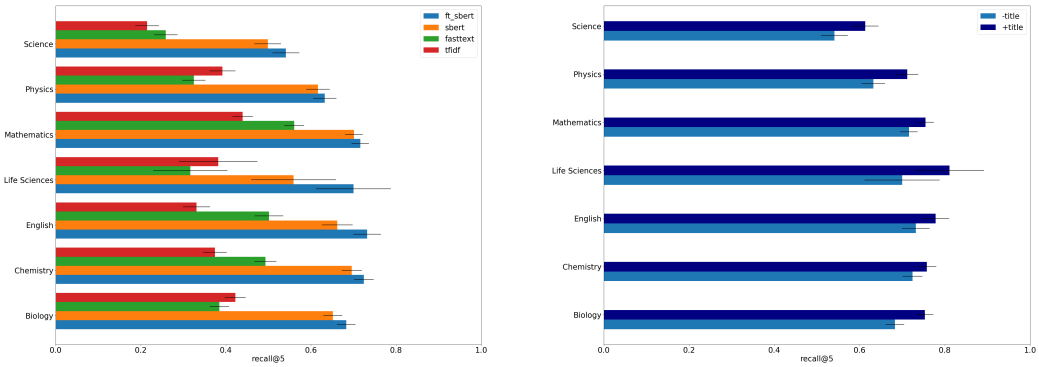
Regarding subject, Figure 6.3a shows that fine-tuned SBERT is the most robust, with the least variation in performance across subjects. Science is the most difficult subject to be matched in all models. This subject is also less standardized than others across curricula, which may render cross-curriculum matching more difficult for this subject.

Note that only the subjects with at least 15 anchor learning objectives were considered. More specific subjects with fewer instances, such "Algebra" from the CCSS curriculum, and "Earth and Space Sciences" from the NGSS curriculum, are left out. Performance is lower for these subjects in general, given that they are less standardized across curricula and usually specific to one curriculum. These subjects originally belong to more specialized curricula, which fill in the subject layer with such niche subjects to fit the tree structure of the collection.

As for the addition of the titles of educational resources to the candidate learning objectives, Figure 6.3a shows that matching improves for all subjects. In general, higher increases are seen for the subjects with lower matching performances, such as Science, Physics and Biology, suggesting a more robust model across subjects.

6.1.4 Curriculum Standardization

Curriculum standardization seems to play a role on cross-curriculum matching. To further verify the effect of cross-curriculum standardization on matching, word n-gram overlap is measured across curricula at the learning objective level. N-gram overlap is a string-based measure of similarity between texts. It may nonetheless provide a quantifiable indication of cross-curriculum heterogeneity. A score of 0 means no overlap between learning objectives and a score of 1 is an exact string match. The lower the n-gram overlap, the higher the heterogeneity, or the lower the standardization.



(a) Recall of each model for each subject.

(b) Recall of fine-tuned SBERT with and without document titles for each subject.

Figure 6.3: Recall@5 for each subject.

I use word unigram (n=1), i.e. lexical, overlap between learning objective pairs. This measure reflects position, that is, common lexicons need to be in the same position in the respective strings to be considered overlap. Also, punctuation and stop words from NLTK are removed and the remaining words are lemmatized to alleviate the limitation of exact matching. Given a pair of curricula, word unigram overlap is measured between each learning objective in curriculum A and each learning objective in curriculum B. For each learning objective in curriculum A, the maximum word unigram overlap is stored and the resulting values are averaged over all learning objectives. First, I check the degree of standardization of learning objectives across curricula, independent of subject or age as measured by lexical overlap (i.e. word unigram overlap). As can be see in Figure 6.4, relatively low lexical overlap is observed across curricula, specially given the pre-processing carried out (e.g. lemmatization). This confirms the high heterogeneity across curricula discussed in section 3.1.

As expected, the CBSE and ICSE curricula have the highest lexical overlap among curriculum pairs. ICSE seems to be the most general curriculum, showing the highest overlap scores with the remaining curricula. Also as expected, Scotland has the lowest lexical overlap with the remaining curricula. These results suggest that curriculum standardization as measured by lexical overlap plays a role in cross-curriculum matching.

Cross-curriculum, lexical overlap is also computed for subjects Science, Mathematics and English, since Science has the lowest matching performance, while Mathematics and English has the highest matching performance. Science shows lower lexical overlap (mean = 0.22) than Mathematics (mean = 0.48) and English (mean = 0.94). When controlling for curriculum by only pairing CBSE and ICSE, the observation still holds: 0.23 for Science, 0.74 for Mathematics and 0.94 for English. Finding a lower overlap for Science than for Mathematics and English suggests that lower standardization in Science may explain, at least in part, the lower matching performance observed for this subject.

Finally, I measure lexical overlap across curricula for ages 11 to 17 to check whether standardization, as measured with word unigram overlap, increases with increasing age.

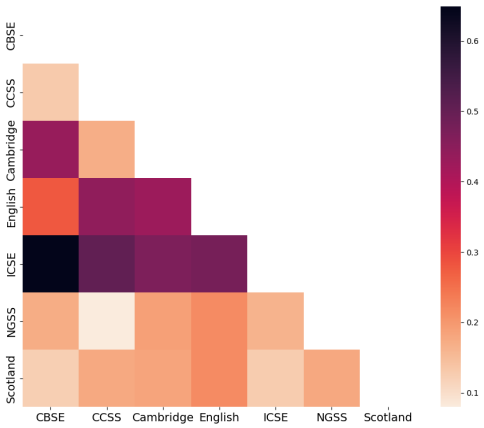


Figure 6.4: Lexical overlap between curricula. The higher the score, the higher the lexical overlap. In general, low overlap is observed across curricula, suggesting high curriculum heterogeneity.

When averaging across curricula and subjects, this is partially observed. Ages 11 and 12 show 0.16 overlap and ages 13 and 14 show 0.24. This increases to 0.28 for age 15, but decreases to 0.21 for age 16, and increases again to 0.51 for age 17. However, note that upon conversation with expert curation, the intuition is that higher ages do have higher standardization in content across curricula. Thus the lack of evidence presented here that supports this idea may also be due to the obvious limitations of word unigram overlap, such as being string-based and not semantic.

In sum, curriculum standardization as measured by lexical overlap seems to affect cross-curriculum matching, at least in relation to curriculum and subject layers. Heterogeneity between learning objectives from certain curricula and subjects is suggested to be associated with matching performance of those learning objectives. That is, the lower the standardization of learning objectives from certain curricula and subjects, the more difficult it is to match them.

6.1.5 Summary

Comparing mean recall@5 across curricula, subjects and ages show that contextualized embeddings, especially from the fine-tuned model, outperform the baselines for all in-domain classes. More specifically, fine-tuned SBERT outperforms the other models independently of subject, age or target curriculum in the data, except for the Scotland curriculum, for which pre-trained SBERT yields the highest score.

Enriching the representation of the candidate learning objectives with fine-grained information from the educational resources increase matching retrieval for all curricula, subjects and ages. In general, increase in performance is seen more significantly for the curricula, subjects and ages with lower matching retrieval, yielding more robust models and suggesting a weakened standardization effect relative to uniquely using the learning objective texts.

It was suggested that curriculum standardization plays a role on matching performance, with higher standardization linked to higher matching scores. This is further

tested by measuring standardization with word unigram overlap between the learning objectives of different curricula, subjects and ages. The results show higher lexical overlap for curricula and subjects with higher matching scores, and vice-versa, further supporting the positive effect of standardization to matching retrieval, at least for curriculum and subject layers.

6.2 Analysis of Hits and Errors

To gain insight into the correct and incorrect matches of each model, match and mismatch types are identified and their proportions computed per model. For this analysis, stratified samples are generated from up to 10% of each model's incorrect matches and correct matches at the highest position in each ranking, with the curricula as strata. The samples varied from 1% to 10% of the total amount of errors or hits of each model. An incorrect match occurs when the model matches two learning objectives that should not be matched, according to the fact that they do not share pinned educational resources. I limit the matching pairs to the highest positioned correct match and incorrect match for each ranking. This is because all learning objectives in the data match at least one other learning objective, but most match fewer other learning objectives than the ranking length (=5), resulting in unavoidable incorrect matches.

A typology of errors and hits was defined by manually checking randomly selected pairs of target and candidate learning objectives and conversing with expert curators. The error types aim at responding why a pair of learning objective incorrectly matched do not correspond to a match. Different levels of the curricula were annotated. Both subject and age levels were labeled as either similar or dissimilar. Similar subjects were either an exact string match or previously mapped as similar subjects, e.g. Algebra to Mathematics. Ages were considered similar when they were the same or differed by a maximum of one year, e.g. 13 and 14. For unit, topic and learning objective levels, the following error types were defined:

- `unrelated`: no overlapping educational content that addresses each item (i.e. unit/topic or learning objective in a pair).
- `related`: some overlapping educational content, but neither is more specific than the other, nor more complex than the other, neither do they take a different approach to the topic.
- `more specific`: the candidate addresses educational content that is too specific for the anchor.
- `more general`: the candidate addresses educational content that is too general for the anchor.
- `more complex`: the candidate addresses a more difficult educational content than the anchor. This can be associated with reading levels, age of the target learner, different stages in the learning trajectory of the curricula, etc.
- `less complex`: the candidate addresses an easier educational content than the anchor. The same factors may be associated as with the *more complex* label, but the direction of complexity is reversed.

- `different approach`: a different approach is taken to the topic, such as for dissimilar purposes, which yields the need for different educational content to address anchor and candidate.
- `same`: anchor and candidate are in fact highly similar, but there is age or subject mismatch, or external factors cause the pair to be labeled as incorrect.

Similarly, the hit types aim at responding why a pair of learning objectives correctly matched indeed correspond to a match. Only one label is given per learning objective pair, and it should correspond to an impression as to why they match, irrespective of the higher layers of the curricula. The hit types are the following:

- `full match`: anchor and candidate have an equivalence relation. All education webpages from the candidate can be used for the anchor and no other educational webpages need to be added.
- `partial match`: anchor and candidate have overlapping meanings. Some educational webpages from the candidate can be used for the anchor, but other educational webpages need to be added still.
- `more specific`: candidate entails a more specific educational content than the anchor. Some educational webpages from the candidate can be used for the anchor, but other educational webpages need to be added still.
- `more general`: candidate entails a more general educational content than the anchor. Some educational webpages from the candidate can be used for the anchor, but other educational webpages need to be added still.

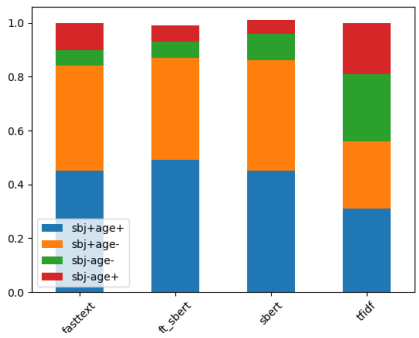
I first pre-annotated the samples using the pre-defined error and hit labels, and then the pre-annotations were later at least partially checked by an expert curator. Section 6.2.1 presents the analysis of errors, and section 6.2.2 presents the analysis of hits.

6.2.1 Incorrect Matching

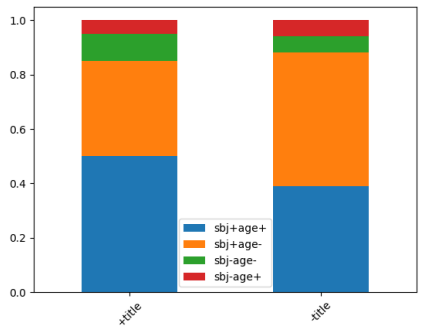
Since age and subject can be annotated automatically for matching, a larger sample with 10% of the errors of each model was used for checking the proportion of errors in which the ages and/or subjects of the wrongly paired learning objectives matched.

As shown in Figure 6.5a, fine-tuned SBERT has the highest proportion of errors with matching subjects and ages across models. In contrast, TF-IDF has the highest proportion of errors with mismatching subjects and ages. The next most common error across models is being in similar subjects but to dissimilar ages. This suggests that the models can, in principle, capture more topicality aspects of the learning objectives, whereas they may fail to capture differences in difficulty level that are associated with the learner age.

When document information is added to the source learning objective, the proportion of errors with matching subjects and ages increase, as can be seen in Figure 6.5b. This is expected, given that fine-grained information is assumed to provide more context to the source learning objective and thus facilitate matching. Differences in age and especially subject are expected to be more easily captured with more detailed information on the learning objective terms. However, more errors with both mismatching subjects and ages is observed. It is possible that some documents expand a wrong



(a) Mismatched subjects and/or ages in errors of each model.



(b) Mismatched subjects and/or ages in errors of fine-tuned SBERT with and without document titles.

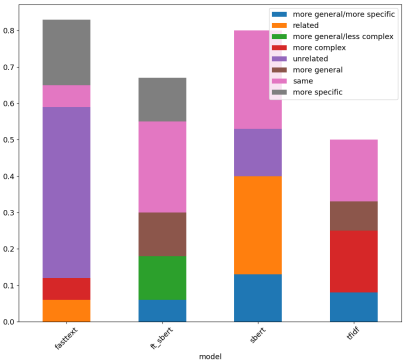
Figure 6.5: Proportion of error types in relation to subject and age. `sbj+age+` corresponds to wrongly paired learning objectives with similar ages and subjects. `sbj+age-` corresponds to wrongly paired learning objectives with similar subjects but dissimilar ages. `sbj-age-` corresponds to wrongly paired learning objectives with dissimilar subjects and ages. `sbj-age+` corresponds to wrongly paired learning objectives with dissimilar subjects but similar ages.

candidate with terms that appear in the anchor, increasing their similarity, while they do not match in the higher-level aspects of subject and age.

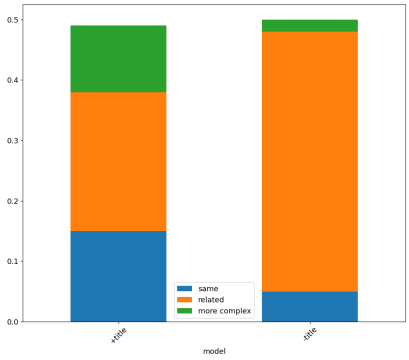
Turning to unit/topic and learning objective layers, Figure 6.6 presents the proportion of error types related to these layers. 1% of all errors of each model (roughly 10 to 20 pairs) was randomly sampled for annotation. Note that an expert curator checked the pre-annotations only partially, thus the results should be considered with upmost caution. In addition, the Figure only includes the combination of error types in topic and query layers which constituted at least 10% of the errors in one of the models.

Figure 6.6a shows the proportion of error types for each encoder using only the learning objective text for matching. Overall, a common error seems to be when both topic and learning objective layers are actually highly similar (labeled `same`). They can be nevertheless considered a mismatch because they differ in subject and/or age. Different subjects could reflect differences in the approach to the educational concepts addressed, and different ages may reflect divergence in complexity level of the educational content. For some instances of this error type, however, the curriculum paths fully matched. The fact that such paired learning objectives do not share pinned educational content may be due to application requirements that are beyond the curriculum path and learning objective texts. For instance, curriculum specifications such as evaluation criteria may affect which education content is curated for a given learning objective, consequently affecting matching. Moreover, some learning objectives in the data appeared to be synthetic queries which did not have any pinned education content and therefore could not have yielded correct matches.

Errors in which both topic and learning objective layers were semantically unrelated do not occur in the sample with TF-IDF and fine-tuned SBERT. Errors in which the predicted candidate is more complex than the anchor seems to occur more often with TF-IDF, indicating difficulties with capturing this difference to avoid mismatches at



(a) Error types for topic and learning objective in each model.



(b) Error types for topic and learning objective in fine-tuned SBERT with or without document titles.

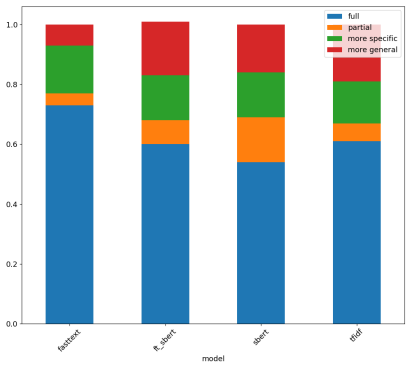
Figure 6.6: Proportion of error types in relation to topic and learning objective.

the topic and learning objective layers. While fine-tuned SBERT does not present any errors from this type, it is the only model to yield errors in which the topic of the candidate is more general than the topic of the anchor, and the candidate is less complex than the anchor. One can imagine that granularity and complexity differences may overlap to some extent, where more general topics may have less complex learning objectives.

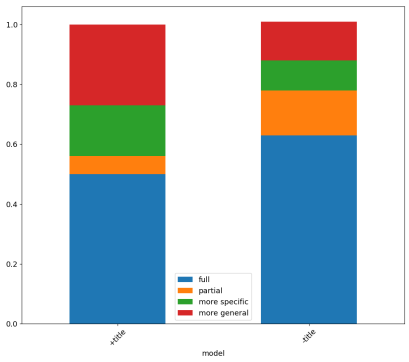
The addition of document titles seems to decrease the errors in which candidate is semantically related to anchor and increase the errors in which candidate and anchor are actually highly similar. As discussed before, being highly similar and yet being a mismatch can be caused by various factors which are not considered in the model, at least explicitly. Thus, these represent the most difficult cases for the model. There is also an increase in errors in which the candidate learning objective and its topic address more complex educational content than the anchor, and this difference could be derived from the topic and learning objectives with the necessary in-domain knowledge. This suggests that the addition of document titles to the candidate learning objectives is not sufficient to provide the model with the needed in-domain knowledge to capture such dissimilarities.

6.2.2 Correct Matching

For attesting the proportion of hit types by each model, I randomly selected 5% of the total amount of correct matches per model. These annotations were not checked by a expert curator, thus they should be considered cautiously. Figure 6.7 shows the proportion of hit types per model. All models seem to mostly generate a full match as the first correct match in a ranking. Figure 6.7a contains the hit types when only the text of the learning objectives is used. Interestingly, fine-tune SBERT and TF-IDF obtained similar proportion of hit types. When document titles are added to the candidates with fine-tuned SBERT, a higher proportion of correct candidates which are more general or more specific than the anchor is observed, as Figure 6.7b shows. This can be interpreted as evidence for the capacity of the model to better capture more nuanced matches.



(a) Hit types in each model.



(b) Hit types in fine-tuned SBERT with or without document titles.

Figure 6.7: Proportion of hit types.

6.2.3 Summary

Checking the proportion of error types for each model showed that all models fail to capture differences in difficulty level between the learning objectives to some extent. Fine-tuned SBERT with document titles presents the highest proportion of errors with matching subjects and ages, suggesting that this model's errors may be more nuanced cases and thus difficult to avoid. This is further supported by the observation that this model also shows the highest proportion of errors in which the candidate and anchor learning objectives and their respective topics are actually highly similar to each other. This type of error can also be considered more nuanced and difficult to avoid. Finally, checking the proportion of hit types indicated that all models mostly generate a full match as the highest correct match in the ranking, and more nuanced matches are captured, i.e. candidate is more general or more specific than anchor, when document titles are added to the candidate learning objective with fine-tuned SBERT.

6.3 Manual Analysis

To gain further insight into the strengths and weaknesses of the best performing model, a manual inspection of a subset of the returned rankings is carried out. Manually inspecting the output allows for anecdotal observations on the strengths and weaknesses of the best performing model, compared to the remaining models.

I first compute the Rank-Biased Overlap (RBO) (Webber et al., 2010) between ranking pairs, to find the most dissimilar rankings across models. Checking the most dissimilar rankings can provide an idea as to how the best performing model outputs differ from the remaining models.

Additionally, I randomly sample 10 anchor learning objectives and respective rankings from each model for each performance combination. The combinations are the following:

- `advantaged`: at least one correct match in the ranking by fine-tuned SBERT, but no correct matches in the rankings by the remaining models for the same anchor

learning objective. This may give an indication of the advantages of fine-tuned SBERT in relation to the remaining models.

- `disadvantaged`: no correct matches in the rankings by fine-tuned SBERT, but at least one correct match in the rankings by pre-trained SBERT, TF-IDF or Fasttext. This may hint to possible disadvantages of fine-tuned SBERT compared to the remaining models.
- `all-bad`: no correct matches in rankings from any model for a given anchor learning objective. This may suggest what is difficult for all models.

The resulting observations are presented in section 6.3.1 for the fine-tuned SBERT with only learning objective texts, and in section 6.3.2 for the fine-tuned SBERT with fine-grained information enriching the representation of the candidate learning objective. The indexing of `CANDIDATE` indicates the position of the candidate learning objective in the top 5. In some cases, the candidates at lower positions in the top k are shown to illustrate an observation. Note that further verification would be needed to draw strong conclusions, such as by probing the model's observed strengths or annotating the rankings with the observed strengths and weaknesses to verify the hypothesis drawn from manual inspection. I leave this to follow-up studies.

6.3.1 Contextualized Representations

Fine-tuned SBERT strengths

Word Sense Disambiguation. Polysemous terms seems to be better captured with contextualized embeddings. 12 illustrates this observation. The predictions from fine-tuned SBERT refer to "roots" and "pressure" in the topic of plant physiology, similarly to the sense of "pressure" in the anchor. Fasttext, on the other hand, predicts "pressure" under the topic of motion and force from the subject physics, and "blood pressure" from the topic of human physiology, both being different, although related, senses of the term "pressure".

12. `ANCHOR`: Root Pressure
    `CANDIDATE 2` fine-tuned SBERT: root pressure and guttation
    `CANDIDATE 2` Fasttext: Pressure
    `CANDIDATE 3` fine-tuned SBERT: Root and Root hair
    `CANDIDATE 3` Fasttext: Blood Pressure

Context-oriented Matching. Semantic similarity is more context-oriented with fine-tuned SBERT. Example 13 illustrates this observation. While TF-IDF suggests learning objectives that either are about plants or breathing, fine-tuned SBERT predicts learning objectives whose meaning includes aspects of the semantics of both lexicons.

13. `ANCHOR`: Do plants breath?
    `CANDIDATE 1` fine-tuned SBERT: Respiration in plants
    `CANDIDATE 1` TF-IDF: Why do we Breathe?
    `CANDIDATE 2` fine-tuned SBERT: Types of respiration in plants
    `CANDIDATE 2` TF-IDF: How do plants grow?
    `CANDIDATE 3` fine-tuned SBERT: Process of respiration in plants
    `CANDIDATE 3` TF-IDF: What nutrients do plants need?

In-domain Meaning. Fine-tuned SBERT seems better at capturing in-domain meaning of terms. This advantage occurs in two directions: with under-specified anchors, the model can match more descriptive candidates (see Examples 14 and 15), while with more descriptive candidates, the model can abstract away and match more general candidates (see Examples 16 and 17).

14. `ANCHOR`: Glycolysis
    CANDIDATE 2 fine-tuned SBERT: respiration in plants glycolysis
    `CANDIDATE TF-IDF`: Glycolysis

15. `ANCHOR`: Resistors in Series and Parallel
    `CANDIDATE 5` fine-tuned SBERT: Combined resistance of two resistors in parallel is less than that of either resistor by itself
    `CANDIDATE` TD-IDF: Series and Parallel circuits
    `CANDIDATE 5` Fasttext: Series and Parallel Circuits - Applications

16. `ANCHOR`: Common Insects- ants, beetles, bees, flies, mosquitoes, butterfly
    `CANDIDATE 1` fine-tuned SBERT: Insects and Creepy crawlies
    `CANDIDATE 1` Fasttext: Microbodies
    `CANDIDATE 1` TF-IDF: Bee extinction

17. `ANCHOR`: Rods and cones in the human retina
    `CANDIDATE 1` fine-tuned SBERT: Structure of Human Eye
    `CANDIDATE 1` Fasttext: Components of the kidney tubules and function
    `CANDIDATE 1` TF-IDF: Frustum of a Cone

Synonyms. The model seems to better capture semantic similar/related words with different word forms. As can be seen in Example 18, the model predicts the correct match by capturing the similarity between "internal" and "detailed" structure. The baseline, on the other hand, incorrectly predicts a related but dissimilar learning objective as it is more general (i.e. structure is not necessary detailed) and includes other aspects (i.e. function).

18. `ANCHOR`: Detailed Structure of Chloroplast
    `CANDIDATE 1` fine-tuned SBERT: Internal Structure of Chloroplast
    `CANDIDATE 1` TF-IDF: Structure and Function of Chloroplast

Fine-tuned SBERT weaknesses

Vagueness due to lack of context. The model is not able to retrieve correct matches in cases where the anchor seems to lack context to accurately determine its meaning. The learning objective text is insufficient to inform the model for semantic matching. Importantly, humans may not perform particularly better in such cases, because information needed to disambiguate is simply not present in the learning objective text. Examples 19 and 20 illustrate this observation.

19. `ANCHOR`: Temperature
    `MATCH A`: Limiting factors in photosynthesis
    `MATCH B`: Factors Affecting Photosynthesis

CANDIDATE 1 fine-tuned SBERT: Measurement of Temperature
CANDIDATE 1 TF-IDF: What is Temperature?

20. ANCHOR: Relative Density
MATCH A: measurement - density of solids
MATCH B: Relative Density of Solid - Archimedes' principle
CANDIDATE 1 fine-tuned SBERT: Density and its formula
CANDIDATE 2 fine-tuned SBERT: Density based on area and volume
CANDIDATE 3 fine-tuned SBERT: definition of population density
CANDIDATE 4 fine-tuned SBERT: Measurement of Density of Liquid

6.3.2 Fine-grained Information

Advantages of enriching candidate representations

Adding fine-grained information from the education content to enrich the representations of candidate learning objectives seem to help the model to predict the correct match for cases in which the correct match lacks context. In Example 21, the correct match is only retrieved when additional information on the correct match, i.e. the titles of the linked educational resources, is used. The titles expand the candidate string with one or more terms that also appear in the anchor. The same occurs in Examples 22 and 23.

21. ANCHOR: ecosystem biodiversity and benefit to humans
MATCH: Healthy Ecosystems
TITLES: Ecosystem Biodiversity, Learning Ecosystems for Kids

22. ANCHOR: Compare lengths, areas and volumes using ratio notation
MATCH A: Area of similar shapes
TITLES A: Similarity and Area Ratios Lengths, areas and volumes of similar shapes
MATCH B: Volume of similar shapes
TITLES B: Similarity and Volume Ratios Lengths, areas and volumes of similar shapes

23. ANCHOR: Biogeochemical Cycles
MATCH: The carbon cycle
TITLES: Biogeochemical Cycles, The Carbon Cycle, Carbon cycle

As can be seen in Examples 24 and 25, titles may repeat the candidate learning objective string, which in turn overlaps with the anchor, increasing the similarity score for that candidate and consequently its position in the ranking.

24. ANCHOR: nature of reactants and products chemical reaction
MATCH: Chemical Reaction
TITLES: What are chemical reactions? What are chemical reactions used for?

25. ANCHOR: Adjectives types
MATCH: Grammar Adjectives Grammar Test
TITLES: Adjectives, How to Use Adjectives, What Are Adjectives?

Disadvantages of enriching candidate representations

In some cases, it appears that the titles do not provide sufficient context to retrieve the correct match or to not retrieve incorrect matches. Specially in the cases in which the anchor is vague, enriching the representations of the candidates may not be sufficient. In Example 26, the match titles do not contain any mention of the terms that appear in the anchor. Meanwhile, the top suggestion by the model is expanded with titles that contain the anchor terms.

26. `ANCHOR`: Contact Forces
    `MATCH`: Friction
    `TITLES`: friction: Factors That Influence Friction, What is friction? Friction
    `CANDIDATE 1`: Contact Forces - Examples
    `TITLES`: Kinds of Forces: Contact Forces, What are frictional forces? Forces

A similar case is seen in 27, where all terms from the anchor, "fertilisation", "sexual" and "reproduction" are seen in the titles of the incorrect match, while only one anchor term "fertilisation" is seen in the titles of the correct match. Additionally, the fact that the anchor refers to human reproduction, as opposed to other types of reproduction, is not possible to infer from the anchor text alone.

27. `ANCHOR`: Fertilisation in sexual reproduction
    `MATCH`: Events during fertilisation
    `TITLES`: fertilization: Events of fertilization, The process of fertilization in human, Fertilization `CANDIDATE 1`: Sexual Reproduction in Plants
    `TITLES`: angiosperm: Reproduction: Fertilization and embryogenesis, Sexual reproduction in plants, Sexual Reproduction In Plant, double fertilization

6.3.3 Summary

Anedoctal observations suggested fine-tuned SBERT to be better at capturing the meaning of expressions, leading to better word sense disambiguation and in-domain semantic similarity. However, further testing is needed to conclude whether this is indeed the case. For instance, by probing the model's capabilities for word sense disambiguation and other meaning distinctions particular to the domain of education, such as differences in complexity and granularity.

Manual inspection further suggested that the model fails when the learning objective text is not sufficiently informative. This weakness is addressed to some extent when the titles of educational resources are added to the candidate representations. However, it is not clear whether the information provided by document segments is indeed more detailed or explanatory information on the meaning of the candidate learning objective. Adding document titles may serve as a term expansion of the candidate, relying more on term overlap than semantics. That is, they seem to merely repeat common terms to anchor and candidate or add new terms to the candidate that are also in the anchor.

Chapter 7

Discussion

Matching taxonomies with semantic similarity in the education domain showed promising results with respect to using transformer-based text encoders and fine-grained information for semantic matching. However, attempts to incorporate information from the higher layers of the curriculum taxonomy were not able capture some aspects beyond semantic similarity needed for accurate matching. This chapter will discuss the main limitations of this study and point out directions for future work.

Domain- and application-specific matching An important challenge in using semantic similarity to match learning objectives across curricula is that being semantically similar may not necessary translate into a correct match between learning objectives, because domain and application-specific factors may affect matching. As mentioned in Section 3.1, curriculum heterogeneity poses a challenge, as learning objectives from different curricula may need distinct educational resources to be achieved depending on other educational or pedagogical factors, such as the age of the target learner. Curriculum specifications, such as the evaluation criteria, also play a role on whether learning objectives from different curricula can be addressed with the same educational documents. A good model for matching learning objectives across curricula must be able to capture and combine such domain- and application-specific factors to semantic similarity.

Adding the texts from the higher layers of the curriculum were an attempt to overcome some of the limitations to semantic similarity of short texts and meet domain and application-specific requirements. In-domain semantics entails nuances on semantic similarity between terms and expressions that are specific to education. The addition of subject and topic labels to the the input with methods based on semantic similarity seemed to have been unsuccessful in fully capturing such in-domain meaning nuances. Similarly, adding grade mapped into age seemed to not not have helped to capture the compatibility in complexity required by the application between education resources which the learning objectives address. Future work may investigate which other methods may be more suitable for capturing and combining domain and application-specific factors with semantic similarity.

Another potential issue with the higher layers of the curriculum is how informative they can be to aid matching with semantic similarity. Some learning objectives annotated as matches were found to target mismatched ages (e.g. 8 and 15). I suggest checking the proportion of the learning objective pairs annotated as matches whose higher layers also match and whether there is a positive correlation between being a

match and having matching higher layers. Some learning objectives annotated as a match were also found to belong to dissimilar topics, as indicated by low cosine. For more textual layers such as unit and topic, I suggest to compare the semantic similarity between learning objectives annotated as matches and the mismatches incorrectly retrieved by the model. No positive correlation between cosines and being a match would at least suggest that semantic similarity in the higher layers may not be a good method to use this information for matching.

Model learning Focusing on model training for matching, a potential issue is the divergence between training and test tasks. During training, the model needs to distinguish between positive and negative examples with likely highly diverging meanings. This is because negative examples are randomly sampled from the remaining learning objectives. In contrast, during test the model must also discard more subtle mismatches to be able to retrieve matches in the ranking top. As a result, high performance on binary classification may be observed during validation, whereas no learning is seen on ranking metrics. Future work may experiment with more informative negative sampling methods.

Task formulation The matching task was formulated as pairwise semantic similarity, which is in turn determined by the cosine between the text encodings of the learning objectives. This approach focus on text representations at the instance or element level, leaving the structure of the curriculum taxonomies aside. Attempts in this study to incorporate information from the higher layers may have failed to improve matching because they were limited to the textual information and did not take into account structural relations. In addition, matching learning objectives was pairwise and independent of each other. This approach cannot explicitly capture dependencies between learning objectives within curriculum, as well as one-to-many or many-to-many mappings between learning objectives.

As literature on ontology alignment suggests, ontology structure may be informative for matching (e.g. Lmati et al. (2015) in the education domain). Combining structure and element-based information for a more holistic ontology alignment is one of the main challenges in the field (Otero-Cerdeira et al., 2015). Future work may investigate how to combine structure information and semantic similarity to aid matching of education curricula. I suggest experimenting with methods that combine semantic similarity with more structure-based measures that check for logic inconsistencies in the predicted matches in the form of matching repair, as proposed by (He et al., 2021). To address the pairwise limitation, a potentially promising venue is to use the stable marriage algorithm (Gale and Shapley, 1962) for many-to-many mappings, applied by Kolyvakis et al. (2018) to ontology alignment, in combination with semantic similarity on contextualized embeddings.

In sum, a good model for matching learning objectives across education curricula requires capturing semantics, but also domain- and application-specific similarity aspects. Representing learning objectives with contextualized embeddings and adding segments from education documents improved matching compared to the baselines, but further testing what information these methods can better capture is still needed. Combining information from higher layers to aid matching did not provide the domain knowledge and application requirements needed, requiring further investigation on whether they are sufficiently informative, and which other approaches may be more suitable

for incorporating this information. Limitations from the task formulation include being pairwise and independent matching. Future work may explore the combination of semantic similarity with contextualized embeddings and structure-based measures in matching education taxonomies.

Chapter 8

Conclusion

This study investigated the task of matching learning objectives across education curricula and proposed a model based on semantic similarity for this task. As such, the focus was on getting representations that could capture in-domain semantic relations and application-specific aspects to generate useful alignments. Comparing contextualized embeddings to static embeddings and to representations based on term-document frequency indicated that contextualized embeddings are better fit to matching learning objectives than the baselines, specially if fine-tuned for the domain and task of interest.

Because the learning objectives in the curricula are extremely short texts, vagueness can be problematic for accurate matching. To address this issue, I tested whether adding the information from higher and lower layers of the curriculum taxonomy in which the learning objectives are embedded further improved matching. The results suggest that information from the lower layer supports matching and provide a more robust model across target curricula. Combining information from the higher layers with semantic similarity, however, did not show any improvement in performance scores, indicating the model was not able to fully capture domain- and application-specific aspects of matching. This result suggests that matching learning objectives seems to require knowledge that goes beyond semantic similarity.

Bibliography

E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, et al. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263, 2015.

A. Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

F. Ardjani, D. Bouchiha, and M. Malki. Ontology-alignment techniques: Survey and analysis. *International Journal of Modern Education & Computer Science*, 7(11), 2015.

P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

A. E. Baker and K. Hengeveld. *Linguistics*. John Wiley & Sons, 2012.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

C. J. Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11 (23-581):81, 2010.

D. Chandrasekaran and V. Mago. Evolution of semantic similarity—a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–37, 2021.

A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

J. Euzenat, P. Shvaiko, et al. *Ontology matching*, volume 18. Springer, 2007.

D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto. The agreementmakerlight ontology matching system. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 527–541. Springer, 2013.

D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

Y. Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420, 2016.

N. Guarino. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, 1998.

N. Guarino and P. Giaretta. Ontologies and knowledge bases. *Towards very large knowledge bases*, pages 1–2, 1995.

M. Han, X. Zhang, X. Yuan, J. Jiang, W. Yun, and C. Gao. A survey on the techniques, applications, and performance of short text semantic similarity. *Concurrency and Computation: Practice and Experience*, 33(5):e5971, 2021.

Y. He, J. Chen, D. Antonyrajah, and I. Horrocks. Bertmap: A bert-based ontology alignment system. *arXiv preprint arXiv:2112.02682*, 2021.

E. Jiménez-Ruiz and B. Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *International Semantic Web Conference*, pages 273–288. Springer, 2011.

D. Jurafsky and J. H. Martin. Speech and language processing,(draft) edition, chapter 6, 2021.

W. A. Kibbe, C. Arze, V. Felix, E. Mitraka, E. Bolton, G. Fu, C. J. Mungall, J. X. Binder, J. Malone, D. Vasant, et al. Disease ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data. *Nucleic acids research*, 43(D1):D1071–D1078, 2015.

P. Kolyvakis, A. Kalousis, and D. Kiritsis. Deepalignment: Unsupervised ontology matching with refined word vectors. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 787–798, 2018.

B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*, 2020.

J. Lin, R. Nogueira, and A. Yates. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325, 2021.

I. Lmati, F. Z. Guerss, M. Aitdaoud, K. Douzi, H. Benlahmar, M. Talbi, N. Achtaich, and A. Namir. Alignment between two domain ontologies (case of educational orientation in mathematics education). In *2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA)*, pages 1–3. IEEE, 2015.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

R. Nogueira and K. Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

L. Otero-Cerdeira, F. J. Rodríguez-Martínez, and A. Gómez-Rodríguez. Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949–971, 2015.

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

A. S. Ramkumar and B. Poorna. Ontology based semantic search: an introduction and a survey of current approaches. In *2014 International Conference on Intelligent Computing Applications*, pages 372–376. IEEE, 2014.

N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

A. Sayed and A. Al Muqrishi. Ibri-casonto: Ontology-based semantic search engine. *Egyptian Informatics Journal*, 18(3):181–192, 2017.

F. Torregrossa, R. Allesiardo, V. Claveau, N. Kooli, and G. Gravier. A survey on training and evaluation of word embeddings. *International Journal of Data Science and Analytics*, pages 1–19, 2021.

A. Usta, I. S. Altingovde, R. Ozcan, and Ö. Ulusoy. Learning to rank for educational search engines. *IEEE Transactions on Learning Technologies*, 14(2):211–225, 2021.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

K. Wang, N. Reimers, and I. Gurevych. Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning. *arXiv preprint arXiv:2104.06979*, 2021.

W. Webber, A. Moffat, and J. Zobel. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38, 2010.

J. Wieting, M. Bansal, K. Gimpel, and K. Livescu. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*, 2015.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Y. Zhang, X. Wang, S. Lai, S. He, K. Liu, J. Zhao, and X. Lv. Ontology matching with word embeddings. In *Chinese computational linguistics and natural language processing based on naturally annotated big data*, pages 34–45. Springer, 2014.