AAST Dataset - Academic Article Survey Tables Documentation

Anonymous

1 Introduction

The Academic Article Survey Table (AAST) dataset was collected using the arXiv and Semantic Scholar APIs. It underwent a series of processing steps, including manual inspection and editing.

The processing steps are summarized as follows:

1) Fetching Survey Papers via the Arxiv API:

2) Preprocessing and Extracting Tables from Latex Files:

3) Creating and Partitioning the Golden Table:

4) Generating Descriptions for Column Headers:

5) Acquiring Citation Data:

2 Provided Files

We offer three files: the main file named "AAST.json", as well as two reference document detail files: "corpus_id_introduction" and "corpus_id_ori_and_sc_other".

- **AAST.json**: This is our primary file, with its content sourced from both the AAST section of the survey paper and the basic details of the references. In addition, we also provide abstracts of the references and fields generated by GPT-4-8k(OpenAI, 2023) for further elucidation.
- **corpus_id_introduction.json**: Utilizing the S2ORC(Ammar et al., 2018) database, this file extracts the introduction section of each reference based on its corpus_id. Both the original and the compressed versions by GPT-3.5-16k are provided for each document.
- **corpus_id_ori_and_sc_other.json**: Leveraging the S2ORC database, this file fetches other sections of the reference papers (excluding any images and tables) based on their corpus_id. Both the original and the GPT-3.5-16k compressed versions are available for each section.

3 Field Description of AAST

“<arxiv_id>”, “<table_id>”, “<row>”, “<column>” are variable placeholders.

```
{
    "id" = "<arxiv_id>-<table_id>-<row>-<column>",

    "arxiv_id" : "arxiv_id" ,
    "table_id" : "table_id (e.g. 0, 1, 1; can mappting with Golden Table)"


```

```
    "chatGPT_field_description": "all supplementary information",
    "chatGPT_field_c_description" : "Column Name (extraction from all supplementary information)",
    "chatGPT_field_cd_description" : "Column Name  + Data Type (extraction from all supplementary information)",
    "chatGPT_field_cdp_description" : "Column Name  + Data Type + Propose (extraction from all supplementary information)",
    "chatGPT_field_cde_description" : "Column Name  + Data Type + Example (extraction from all supplementary information)",
    "chatGPT_field_cdpe_description" : "Column Name  + Data Type + Propose + Example (extraction from all supplementary information)",


    "ref_title": "The Title of the referenced literature.",
    "ref_abstract": "Abstract of the referenced literature.",
    "corpus_id": "ID of the reference in S2ORC of Semantic Scholar. You can use this corpus id to search detail information at S2ROC dataset or S2AG API. 'https://github.com/allenai/s2orc' and 'https://www.semanticscholar.org/product/api'",

    "ref_paper_id":"paper ID of the reference in Semantic Scholar",

    "ref_year" : "Publication year of the referenced literature.",
    "ref_authors": "Author list of the referenced literature.",
    "ref_apa" : "The generated in-text citation in APA format.",
    "fulltext": "Fulltext of paper",
    "gold_header_name_and_answer_pair_tsv": "A TSV format containing table headers and answers: the first row represents the table headers, while the second row contains the answers.",
    "gold_header_name_and_answer_pair_json": "A JSON format where the table header serves as the key and its corresponding answer as the value."

}
```

3.1 AAST Example

```

{
    "id": "1904.05046v3-2-5-1",
    "arxiv_id": "1904.05046v3",
    "table_id": "2",
    "paper_title": "Generalizing from a Few Examples: A Survey on Few-Shot Learning",
    "table_title": "Table 5. Characteristics of embedding learning methods.",
    "chatGPT_field_description": "Column Name: category\nData Type: String\nPurpose: This column indicates the type of method used for embedding learning, which can be task-specific, task-invariant, or hybrid. Task-specific methods are designed for a particular task, while task-invariant methods can be applied to various tasks. Hybrid methods combine elements of both task-specific and task-invariant approaches.\nExample: task-specific, task-invariant, hybrid\n\nColumn Name: method\nData Type: String\nPurpose: This column lists the name of the embedding learning method along with the author(s) and publication year. These methods are used to
```

```
    learn representations of data in a lower-dimensional space, which can be
    useful for tasks such as classification, clustering, or similarity search
    .\nExample: Method A (Author et al., Year), Method B (Author, Year),
    Method C (Author, Year)\n\nColumn Name: embedding function f for x test\
    nData Type: String\nPurpose: This column describes the function used to
    generate embeddings for the test data. The function can be a type of
    neural network (e.g., CNN, LSTM, GNN) or another method (e.g., kernel,
    logistic projection, adaptive CNN).\nExample: Neural Network A, Method D,
    Method E\n\nColumn Name: embedding function g for D train\nData Type:
    String\nPurpose: This column describes the function used to generate
    embeddings for the training data. The function can be the same as the one
    used for test data or a different one (e.g., biLSTM, another CNN).\
    nExample: the same as f, Neural Network B, Neural Network C\n\nColumn Name
    : similarity measure s\nData Type: String\nPurpose: This column specifies
    the similarity measure used to compare embeddings. The similarity measure
    can be a distance metric (e.g., Distance A, Distance B, Distance C) or
    another method (e.g., Similarity D, Similarity E, Learned Distance).\
    nExample: Similarity D, Distance A, Distance B\n\nTable: Method",
"chatGPT_field_c_description": "Column Name: category\n\nColumn Name: method\n
    \nColumn Name: embedding function f for x test\n\nColumn Name: embedding
    function g for D train\n\nColumn Name: similarity measure s",
"chatGPT_field_cd_description": "Column Name: category\nData Type: String\n\
    nColumn Name: method\nData Type: String\n\nColumn Name: embedding function
     f for x test\nData Type: String\n\nColumn Name: embedding function g for
    D train\nData Type: String\n\nColumn Name: similarity measure s\nData Type
    : String",
"chatGPT_field_cdp_description": "Column Name: category\nData Type: String\
    nPurpose: This column indicates the type of method used for embedding
    learning, which can be task-specific, task-invariant, or hybrid. Task-
    specific methods are designed for a particular task, while task-invariant
    methods can be applied to various tasks. Hybrid methods combine elements
    of both task-specific and task-invariant approaches.\n\nColumn Name:
    method\nData Type: String\nPurpose: This column lists the name of the
    embedding learning method along with the author(s) and publication year.
    These methods are used to learn representations of data in a lower-
    dimensional space, which can be useful for tasks such as classification,
    clustering, or similarity search.\n\nColumn Name: embedding function f for
     x test\nData Type: String\nPurpose: This column describes the function
    used to generate embeddings for the test data. The function can be a type
    of neural network (e.g., CNN, LSTM, GNN) or another method (e.g., kernel,
    logistic projection, adaptive CNN).\n\nColumn Name: embedding function g
    for D train\nData Type: String\nPurpose: This column describes the
    function used to generate embeddings for the training data. The function
    can be the same as the one used for test data or a different one (e.g.,
    biLSTM, another CNN).\n\nColumn Name: similarity measure s\nData Type:
    String\nPurpose: This column specifies the similarity measure used to
    compare embeddings. The similarity measure can be a distance metric (e.g.,
     Distance A, Distance B, Distance C) or another method (e.g., Similarity D
    , Similarity E, Learned Distance).",
"chatGPT_field_cde_description": "Column Name: category\nData Type: String\
    nExample: task-specific, task-invariant, hybrid\n\nColumn Name: method\
    nData Type: String\nExample: Method A (Author et al., Year), Method B (
    Author, Year), Method C (Author, Year)\n\nColumn Name: embedding function
    f for x test\nData Type: String\nExample: Neural Network A, Method D,
```

```
        Method E\n\nColumn Name: embedding function g for D train\nData Type:
        String\nExample: the same as f, Neural Network B, Neural Network C\n\
        nColumn Name: similarity measure s\nData Type: String\nExample: Similarity
         D, Distance A, Distance B",
    "chatGPT_field_cdpe_description": "Column Name: category\nData Type: String\
        nPurpose: This column indicates the type of method used for embedding
        learning, which can be task-specific, task-invariant, or hybrid. Task-
        specific methods are designed for a particular task, while task-invariant
        methods can be applied to various tasks. Hybrid methods combine elements
        of both task-specific and task-invariant approaches.\nExample: task-
        specific, task-invariant, hybrid\n\nColumn Name: method\nData Type: String
        \nPurpose: This column lists the name of the embedding learning method
        along with the author(s) and publication year. These methods are used to
        learn representations of data in a lower-dimensional space, which can be
        useful for tasks such as classification, clustering, or similarity search
        .\nExample: Method A (Author et al., Year), Method B (Author, Year),
        Method C (Author, Year)\n\nColumn Name: embedding function f for x test\
        nData Type: String\nPurpose: This column describes the function used to
        generate embeddings for the test data. The function can be a type of
        neural network (e.g., CNN, LSTM, GNN) or another method (e.g., kernel,
        logistic projection, adaptive CNN).\nExample: Neural Network A, Method D,
        Method E\n\nColumn Name: embedding function g for D train\nData Type:
        String\nPurpose: This column describes the function used to generate
        embeddings for the training data. The function can be the same as the one
        used for test data or a different one (e.g., biLSTM, another CNN).\
        nExample: the same as f, Neural Network B, Neural Network C\n\nColumn Name
        : similarity measure s\nData Type: String\nPurpose: This column specifies
        the similarity measure used to compare embeddings. The similarity measure
        can be a distance metric (e.g., Distance A, Distance B, Distance C) or
        another method (e.g., Similarity D, Similarity E, Learned Distance).\
        nExample: Similarity D, Distance A, Distance B",
    "ref_title": "Matching networks for one shot learning",
    "ref_abstract": "Learning from a few examples remains a key challenge in
        machine learning. Despite recent advances in important domains such as
        vision and language, the standard supervised deep learning paradigm does
        not offer a satisfactory solution for learning new concepts rapidly from
        little data. In this work, we employ ideas from metric learning based on
        deep neural features and from recent advances that augment neural networks
         with external memories. Our framework learns a network that maps a small
        labelled support set and an unlabelled example to its label, obviating the
         need for fine-tuning to adapt to new class types. We then define one-shot
         learning problems on vision (using Omniglot, ImageNet) and language tasks
        . Our algorithm improves one-shot accuracy on ImageNet from 87.6% to 93.2%
         and from 88.0% to 93.8% on Omniglot compared to competing approaches. We
        also demonstrate the usefulness of the same model on language modeling by
        introducing a one-shot task on the Penn Treebank.",
    "corpus_id": "8909022",
    "ref_paper_id": "be1bb4e4aa1fcf70281b4bd24d8cd31c04864bb6",
    "ref_year": "2016",
    "ref_authors": "Oriol Vinyals;C. Blundell;T. Lillicrap;K. Kavukcuoglu;Daan
        Wierstra",
    "ref_apa": "Vinyals et al., (2016)",
    "fulltext": "...",
```

```
    "gold_header_name_and_answer_pair_tsv": "\tcategory\tembedding function f for
        x test\tembedding function g for D train\tsimilarity measure s\r\n4\ttask-
        invariant\tCNN, LSTM\tCNN, biLSTM\tcosine similarity\r\n",
    "gold_header_name_and_answer_pair_json": [
        {
            "category": "task-invariant",
            "embedding function f for x test": "CNN, LSTM",
            "embedding function g for D train": "CNN, biLSTM",
            "similarity measure s": "cosine similarity"
        }
    ]
},
```

4 Promppt for Semantic Compression

Given the input context limitations of LLMs, it’s impractical to embed an entire document as reference context within the prompt. For this reason, we adopted the "Semantic Compression With Large Language Models" method proposed by Gilbert et al. (2023) and others, and applied it to condense the academic papers. Due to some introductions in the dataset exceeding 8k tokens and considering cost implications, we opted to use GPT 3.5 16K as our model for semantic compression. Furthermore, according to their research, chatGPT 3.5 also exhibits commendable semantic compression capabilities. Utilizing the section structural information provided by the S2ORC dataset, we performed semantic compression on the text of each section.

System prompt:

```
You are a ChatGPT LLM trained by OpenAI to compress text.
The compression model should purely minimize the number of characters in the
    compressed representation, while maintaining the semantics of the original
    text and preserving named entities.
The resulting compressed text does not need to be decompressed into exactly the
    original text, but should capture the semantics of the original text.
The compressed text should be able to be decompressed into a text that is
    semantically similar to the original text, but does not need to be identical.
```

Action Prompt:

“<text>” represents the text that needs to be compressed.

```
[Prompt]
Compress the following text. Return only the compressed
text with no additional text. Text to compress:
[Following Text]
<text>
```

5 Field Description of corpus_id_introduction

The file “corpus_id_introduction.json” primarily provides the Original and Semantic Compression Introduction of References. The format is defined as follows:

```
/*
    "corpus_id" is a variable placeholder: ": "ID of the reference in Semantic Scholar. You can use this corpus id to search detail information at S2ROC dataset or S2AG API. 'https://github.com/allenai/s2orc' and 'https://www.semanticscholar.org/product/api'",
*/
{
    "corpus_id": {
        "ori_introduction": "original introduction context"
        "sc_introduction": "Semantic Compression introduction context"
    }
}
```

Example

```
{
    "8909022": {
        "ori_introduction": "Introduction: \nHumans learn new concepts with very little supervision -e.g. a child can generalize the concept of \"giraffe\" from a single picture in a book -yet our best deep learning systems need hundreds or thousands of examples. This motivates the setting we are interested in: \"one-shot\" learning, which consists of learning a class from a single labelled example.\n\nDeep learning has made major advances in areas such as speech [7], vision [13] and language [16], but is notorious for requiring large datasets. Data augmentation and regularization techniques alleviate overfitting in low data regimes, but do not solve it. Furthermore, learning is still slow and based on large datasets, requiring many weight updates using stochastic gradient descent. This, in our view, is mostly due to the parametric aspect of the model, in which training examples need to be slowly learnt by the model into its parameters.\n\nIn contrast, many non-parametric models allow novel examples to be rapidly assimilated, whilst not suffering from catastrophic forgetting. Some models in this family (e.g., nearest neighbors) do not require any training but performance depends on the chosen metric [1]. Previous work on metric learning in non-parametric setups [18] has been influential on our model, and we aim to incorporate the best characteristics from both parametric and non-parametric models -namely, rapid acquisition of new examples while providing excellent generalisation from common examples.\n\nThe novelty of our work is twofold: at the modeling level, and at the training procedure. We propose Matching Nets (MN), a neural network which uses recent advances in attention and memory that enable rapid learning. Secondly, our training procedure is based on a simple machine learning principle: test and train conditions must match. Thus to train our network to do rapid learning, we Besides our contributions in defining a model and training criterion amenable for one-shot learning, we contribute by the definition of tasks that can be used to benchmark other approaches on both ImageNet and small scale language modeling. We hope that our results will encourage others to work on this challenging problem.\n\nWe organized the paper by first defining and explaining our model whilst linking its several components to related work. Then in the following section we briefly
```

```
            elaborate on some of the related work to the task and our model. In
            Section 4 we describe both our general setup and the experiments we
            performed, demonstrating strong results on one-shot learning on a
            variety of tasks and setups.",
        "sc_introduction": "Introduction: Humans learn new concepts with minimal
            supervision - e.g. a child generalizes the concept of \"giraffe\" from
             a single picture in a book - yet deep learning systems require
            numerous examples. This motivates \"one-shot\" learning, which
            involves learning a class from a single labeled example.\n\nDeep
            learning has made advances in speech [7], vision [13], and language
            [16], but requires large datasets. Data augmentation and
            regularization alleviate overfitting in low data regimes but do not
            solve it. Learning is slow and based on large datasets, requiring many
             weight updates using stochastic gradient descent. This is mainly due
            to the parametric aspect of the model, where training examples need to
             be slowly learned by the model into its parameters.\n\nIn contrast,
            non-parametric models allow rapid assimilation of novel examples
            without catastrophic forgetting. Some models in this family (e.g.,
            nearest neighbors) do not require training but performance depends on
            the chosen metric [1]. Previous work on metric learning in non-
            parametric setups [18] has influenced our model, and we aim to
            incorporate the best characteristics from both parametric and non-
            parametric models - rapid acquisition of new examples while providing
            excellent generalization from common examples.\n\nOur work is novel in
             two ways: at the modeling level and the training procedure. We
            propose Matching Nets (MN), a neural network that uses recent advances
             in attention and memory for rapid learning. Our training procedure is
             based on the principle that test and train conditions must match.
            Thus, to train our network for rapid learning, we Besides defining a
            model and training criterion for one-shot learning, we contribute by
            defining benchmark tasks for ImageNet and small-scale language
            modeling. We hope our results will encourage others to work on this
            challenging problem.\n\nWe organize the paper by first defining and
            explaining our model, linking its components to related work. Then, we
             briefly elaborate on related work to the task and our model. In
            Section 4, we describe our general setup and experiments,
            demonstrating strong results on one-shot learning across various tasks
             and setups."
}
```

6 Field Description of corpus_id_ori_and_sc_other

The file “corpus_id_ori_and_sc_other.json” primarily provides the Original and Semantic Compression Introduction of References. The format is defined as follows:

```
/*
    "<corpus_id>" is a variable placeholder: ": "ID of the reference in Semantic
        Scholar. You can use this corpus id to search detail information at S2ROC
        dataset or S2AG API. 'https://github.com/allenai/s2orc' and 'https://www.
        semanticscholar.org/product/api'",
     "<section_name>" is a variable placeholder
 */
{
```

```
    "<corpus_id>": {
        "<section_name>": {
            "ori_section": "original section context"
            "sc_section": "Semantic Compression section context"
        }, ...
    },...
}
```

Example

```
{
    "8909022": {
        "model": {
            "ori_section": "Our non-parametric approach to solving one-shot
                learning is based on two components which we describe in the
                following subsections. First, our model architecture follows
                recent advances in neural networks augmented with memory (as
                discussed in Section 3). Given a (small) support set S, our model
                defines a function c S (or classifier) for each S, i.e. a mapping
                S\rightarrowcS(.). Second, we employ a training strategy which is
                tailored for one-shot learning from the support set S.",
            "sc_section": "Our non-parametric approach to one-shot learning is
                based on two components: model architecture and training strategy
                ."
        },
        "model architecture": {
            "ori_section": "In recent years, many groups have investigated ways to
                 augment neural network architectures with external memories and
                other components that make them more \"computer-like\". We draw
                inspiration from models such as sequence to sequence (seq2seq)
                with attention [2], memory networks [29] and pointer networks
                [27].\n\nIn all these models, a neural attention mechanism, often
                fully differentiable, is defined to access (or read) a memory
                matrix which stores useful information to solve the task at hand.
                Typical uses of this include machine translation, speech
                recognition, or question answering. More generally, these
                architectures model P (B|A) where A and/or B can be a sequence (
                like in seq2seq models), or, more interestingly for us, a set
                [26].\n\nOur contribution is to cast the problem of one-shot
                learning within the set-to-set framework [26].\n\nThe key point is
                 that when trained, Matching Networks are able to produce sensible
                 test labels for unobserved classes without any changes to the
                network. More precisely, we wish to map from a (small) support set
                 of k examples of image-label pairs S = {(x i , y i )} k i=1 to a
                classifier c S (x) which, given a test examplex, defines a
                probability distribution over outputs\hat{y}. We define the
                mapping S \rightarrow c S (x) to be P (\hat{y}|x, S) where P is
                parameterised by a neural network. Thus, when given a new support
                set of examples S from which to one-shot learn, we simply use the
                parametric neural network defined by P to make predictions about
                the appropriate label\hat{y} for each test examplex: P (\hat{y}|x,
                 S ). In general, our predicted output class for a given input
                unseen examplex and a support set S becomes arg max y P (y|x, S).\
```

```
n\nOur model in its simplest form computes\hat{y} as follows:\ny =
 k i=1 a(x, x i )y i(1)\nwhere x i , y i are the samples and
labels from the support set\nS = {(x i , y i )} k i=1\n, and a is
an attention mechanism which we discuss below. Note that eq. 1
essentially describes the output for a new class as a linear
combination of the labels in the support set. Where the attention
mechanism a is a kernel on X \times X, then (1) is akin to a
kernel density estimator. Where the attention mechanism is zero
for the b furthest x i fromx according to some distance metric and
 an appropriate constant otherwise, then (1) is equivalent to 'k \
minus b'-nearest neighbours (although this requires an extension
to the attention mechanism that we describe in Section 2.1.2).
Thus (1) subsumes ...",
"sc_section": "In recent years, groups have explored ways to enhance
neural network architectures with external memories and components
 to make them more \"computer-like\". We draw inspiration from
models like seq2seq with attention [2], memory networks [29], and
pointer networks [27].\n\nIn these models, a neural attention
mechanism is used to access a memory matrix that stores useful
information for solving tasks. This includes machine translation,
speech recognition, and question answering. These architectures
model P(B|A), where A and/or B can be a sequence or a set [26].\n\
nOur contribution is to approach one-shot learning within the set-
to-set framework [26].\n\nThe key point is that Matching Networks
can produce sensible test labels for unseen classes without
modifying the network. Specifically, we aim to map a support set S
 = {(x_i, y_i)} of k image-label pairs to a classifier c_S(x) that
 defines a probability distribution over outputs \hat{y} for a
test example x. The mapping S \rightarrow c_S(x) is parameterized
by a neural network P. Thus, when given a new support set S for
one-shot learning, we use the neural network P to predict the
appropriate label \hat{y} for each test example x: P(\hat{y}|x, S)
. In general, our predicted output class for an unseen input
example x and support set S is arg max y P(y|x, S).\n\nOur model
computes \hat{y} as follows:\ny = \Sigma a(x, x_i) * y_i (1)\
nwhere x_i, y_i are the samples and labels from the support set S
= {(x_i, y_i)} and a is an attention mechanism. Note that eq. 1
describes the output for a new class as a linear combination of
the labels in the support set. If the attention mechanism a is a
kernel on X \times X, then (1) is similar to a kernel density
estimator. If the attention mechanism is zero for the b furthest
x_i from x according to some distance metric and an appropriate
constant otherwise, then (1) is equivalent to 'k - b'-nearest
neighbors. Thus, (1) encompasses both KDE and kNN methods. Another
 interpretation of (1) is that a acts as an attention mechanism
and y_i act as memories associated with x_i. In this case, it can
be seen as a specific type of associative memory where, given an
input, we \"point\" to the corresponding example in the support
set and retrieve its label. However, unlike other attentional
memory mechanisms [2], (1) is non-parametric: as the support set
size increases, so does the memory used. Therefore, the functional
 form defined by the classifier c_S(x) is highly flexible and can
easily adapt to any new support set."
},
```

```
        "the attention kernel": {
            "ori_section": "Equation 1 relies on choosing a(., .), the attention mechanism, which fully specifies the classifier. The simplest form that this takes (and which has very tight relationships with common attention models and kernel functions) is to use the softmax over the cosine distance c, i.e., a(x, x i ) = e c(f (x),g(xi)) / k j=1 e c(f (x),g(xj )) with embedding functions f and g being appropriate neural networks (potentially with f = g) to embedx and x i . In our experiments we shall see examples where f and g are parameterised variously as deep convolutional networks for image tasks (as in VGG [22] or Inception [24]) or a simple form word embedding for language tasks (see Section 4).\n\nWe note that, though related to metric learning, the classifier defined by Equation 1 is discriminative. For a given support set S and sample to classifyx, it is enough forx to be sufficiently aligned with pairs (x , y ) \in S such that y = y and misaligned with the rest. This kind of loss is also related to methods such as Neighborhood Component Analysis (NCA) [18], triplet loss [9] or large margin nearest neighbor [28].\n\nHowever, the objective that we are trying to optimize is precisely aligned with multi-way, one-shot classification, and thus we expect it to perform better than its counterparts. Additionally, the loss is simple and differentiable so that one can find the optimal parameters in an \"end-to-end\" fashion.",
            "sc_section": "Equation 1 relies on choosing a(., .), the attention mechanism, which fully specifies the classifier. The simplest form : a(x, xi) = e c(f(x),g(xi))/k j=1 e c(f(x),g(xj)) with embedding functions f and g being appropriate neural networks (potentially with f = g) to embed x and xi. In our experiments we shall see examples where f and g are parameterised variously as deep convolutional networks for image tasks (as in VGG [22] or Inception [24]) or a simple form word embedding for language tasks (see Section 4).\n\nWe note that, though related to metric learning, the classifier defined by Equation 1 is discriminative. For a given support set S and sample to classify x, it is enough for x to be sufficiently aligned with pairs (x, y) \in S such that y = y and misaligned with the rest. This kind of loss is also related to methods such as Neighborhood Component Analysis (NCA) [18], triplet loss [9] or large margin nearest neighbor [28].\n\nHowever, the objective we are trying to optimize is precisely aligned with multi-way, one-shot classification, and thus we expect it to perform better than its counterparts. Additionally, the loss is simple and differentiable so that one can find the optimal parameters in an \"end-to-end\" fashion."
        },
        "full context embeddings": {
            "ori_section": "The main novelty of our model lies in reinterpreting a well studied framework (neural networks with external memories) to do one-shot learning. Closely related to metric learning, the embedding functions f and g act as a lift to feature space X to achieve maximum accuracy through the classification function described in eq. 1.\n\nDespite the fact that the classification strategy is fully conditioned on the whole support set through P(.|x, S), the embeddings on which we apply the cosine similarity
```

```
                to \"attend\", \"point\" or simply compute the nearest neighbor
                are myopic in the sense that each element x i gets embedded by g(x
                 i ) independently of other elements in the support set S.
                Furthermore, S should be able to modify how we embed the test
                imagex through f .\n\nWe propose embedding the elements of the set
                 through a function which takes as input the full set S in
                addition to x i , i.e. g becomes g(x i , S). Thus, as a function
                of the whole support set S, g can modify how to embed x i . This
                could be useful when some element x j is very close to x i , in
                which case it may be beneficial to change the function with which
                we embed x i -some evidence of this is discussed in Section 4. We
                use a bidirectional Long-Short Term Memory (LSTM) [8] to encode x
                i in the context of the support set S, considered as a sequence (
                see appendix for a more precise definition).\n\nThe second issue
                can be fixed via an LSTM with read-attention over the whole set S,
                 whose inputs are equal to x:\nf (x, S) = attLSTM(f (x), g(S), K)
                where f (x)\nare the features (e.g., derived from a CNN) which are
                 input to the LSTM (constant at each time step). K is the fixed
                number of unrolling steps of the LSTM, and g(S) is the set over
                which we attend, embedded with g. This allows for the model to
                potentially ignore some elements in the support set S, and adds \"
                depth\" to the computation of attention (see appendix for more
                details).",
            "sc_section": "The main novelty of our model is reinterpreting a well-
                studied framework (NNs with external memories) for one-shot
                learning. Related to metric learning, the embedding functions f
                and g lift features to space X for maximum accuracy in the
                classification function (eq. 1).\n\nDespite the classification
                strategy being conditioned on the support set through P(.|x, S),
                the embeddings are myopic. Each element xi is embedded
                independently by g(xi) regardless of other elements in S.
                Additionally, S can modify how we embed the test image x through f
                .\n\nWe propose embedding the set elements using a function that
                takes S and xi as input, i.e., g becomes g(xi, S). Thus, g can
                modify how xi is embedded based on the support set S. This is
                useful when xi is close to xj, allowing us to change the embedding
                 function for xi. We use a bidirectional LSTM [8] to encode xi in
                the context of S, treated as a sequence.\n\nThe second issue is
                addressed by an LSTM with read-attention over S, where the inputs
                are x:\nf(x, S) = attLSTM(f(x), g(S), K), where f(x) are the
                features (e.g., from a CNN) input to the LSTM (constant at each
                time step). K is the fixed number of unrolling steps of the LSTM,
                and g(S) is the attended set embedded with g. This allows the
                model to potentially ignore some elements in S and adds \"depth\"
                to attention computation."
        },..
    }
    ,...
}
```

References

Ammar, W., Groeneveld, D., Bhagavatula, C., Beltagy, I., Crawford, M., Downey, D., Dunkelberger, J., Elgohary, A., Feldman, S., Ha, V., Kinney, R., Kohlmeier, S., Lo, K., Murray, T., Ooi, H.-H., Peters, M., Power, J., Skjonsberg, S., Wang, L. L., Wilhelm, C., Yuan, Z., van Zuylen, M., and Etzioni, O. (2018). Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 84–91, New Orleans - Louisiana. Association for Computational Linguistics.

Gilbert, H., Sandborn, M., Schmidt, D. C., Spencer-Smith, J., and White, J. (2023). Semantic compression with large language models. *ArXiv*, abs/2304.12512.

OpenAI (2023). Gpt-4 technical report.