Marcus Tiedemann Økland Henriksbø

Prompting generative models for named entity recognition using language and visuals

Master's thesis in Computer Science
Supervisor: Ole Jakob Mengshoel
Co-supervisor: Tong Yu

June 2023

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Marcus Tiedemann Økland Henriksbø

Prompting generative models for named entity recognition using language and visuals

Master's thesis in Computer Science
Supervisor: Ole Jakob Mengshoel
Co-supervisor: Tong Yu
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

The advances in large language models have been noticeable to researchers and the general public in recent times [56]. We see the development of large language models in conjunction with vision models and multimodal models becoming an exciting research space because of the ability to process more information from additional modalities.

In our work, we leverage publicly available large language and multimodal models for the quite established natural language processing task of named entity recognition. Using novel approaches, we explore adapting the task to generative models through prompting to leverage the capabilities of such models without modifying the architecture or training weights. Further, we use the multimodal models with multimodal named entity recognition datasets to experiment with the modelâs ability to leverage visuals for better performance.

Our approach fits into the larger prompting trend but leverages it for a traditional classification task through a novel approach using question-answering prompting with generative models, demonstrating state-of-the-art or competitive performance in the case of zero or very few training examples.

Sammendrag (Norwegian Abstract)

Fremskrittet til store språkmodeller har vært merkbare ikke bare for forskere, men også for allmennheten i nyere tid [56]. Vi ser utviklingen av store språkmodeller i sammenheng med visuelle modeller slik at multimodale modeller kan være interessante for forskning fordi man kan håndtere mer informasjon når man bruker flere typer data.

I vårt arbeid prøver vi å utnytte både noen offentlig tilgjengelige store språkmodeller og multimodale modeller for den ganske etablerte språkbehandlingsoppgaven navngitt entitetsgjenkjenning. Ved å bruke nye tilnærminger utforsker vi hvordan tilpasse oppgaven til generative modeller via spørsmål, for å kunne utnytte mulighetene til slike modeller uten å måtte endre arkitekturen eller trene vekter. Videre utnytter vi de multimodale modellene i forbindelse med multimodale datasett for navngitt entitetsgjenkjenning for å eksperimentere med modellens evne til å utnytte visuelle elementer for bedre ytelse.

Vår tilnærming passer inn i den større spørringstrenden, men utnytter den for en tradisjonell oppgave gjennom en ny tilnærming som bruker spørsmåls-svar med generative modeller, og viser state-of-the-art eller høy ytelse i tilfeller med null eller svært få treningseksempler.

Preface

This master's thesis was conducted at the Norwegian University of Science and Technology in the spring of 2023. This work was written by Marcus Tiedemann Oekland Henriksboe, a master's student at the Department of Computer and Information Science as part of graduating from the program. I wish to thank my supervisor Ole Jakob Mengshoel for his valuable insights and suggestions for research articles. I also want to thank my co-advisor, Dr. Yu Tong for his valuable insight. The topic of the project and its orientation of it toward question-answering approaches were positively influenced by him. Lastly, I want to be thankful for the recent advancements and awareness regarding AI, making it even more exciting to work with.

Marcus Tiedemann Oekland Henriksboe
Trondheim, June 26, 2023

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter explains the conceptual background for the work before introducing our motivation. Further, it proposes research questions and presents the thesis structure.

1.1 Named entity recognition

Named entity recognition (NER) can be segmented into two sub-tasks. First identifying the entities in the text, and then classifying them into some classes, normally predefined.

The CoNLL-2003 format uses four entity classes, person, location, organization, and miscellaneous, in addition to the negative/none class. The miscellaneous is a grouping of all the entities which do not correspond to the first three classes.

Some examples of entities are: ***John Hancock*** was a **person**, **Apple** is an **organization**, **Times Square** is a **location** and **Wednesday** is a **misceallanous entity**.

In a single example, we might structure it like this with the labels marking the entities: [**David Carmack, Person**] who works at [**Starbucks, Organization**] in [**Seattle, Location**] is looking forward to the [**Super Bowl, Miscellaneous**].

1.1.1 Multimodal

In the case of named entity recognition, multimodal has meant including images as an additional visual input to text [86, 77]. In the work by Zhang et al. [105] and Lu et al. [53], this consists of social media posts that have both images and text. The idea is that there is information in the image which is not present in the text itself and that by including the image, the model can better understand the data and classify the entities correctly.

1.2 Motivation

Deep learning has had a great development in the last few years. As large visual-language models become increasingly available, the potential for leveraging multiple modalities becomes more of an accessible and interesting research topic.

In addition, larger models have opened up a new world of few-shot learning at inference, often being able to adapt to new tasks with very few examples. This combination of being able to handle different modalities and quickly adapt to a few examples can make for a potentially very powerful ability to learn a range of new tasks very quickly.

Therefore it is interesting to examine how these abilities can be used for not just ordinary language and image processing and generation, but applicability for narrower classification tasks. Named entity recognition is a relatively traditional NLP task that recently has been combined with utilizing multiple modalities and also being able to use pre-trained models for few-shot learning. We, therefore, are interested in exploring the ability of generative and large visual-language models to deal with multimodal named entity recognition in a few-shot learning context.

Most multimodal models [1, 45, 51, 39] are benchmarked across a range of multimodal benchmarks. As of writing, we have not found other work which leverages these types of generalizable vision-language models for multimodal named entity recognition. This, therefore, presents an opportunity for novel experiments with regards to finding another performant way to do low-resource/few-shot unimodal or multimodal named entity recognition and benchmarking the ability of these generalizable vision and language models in their ability to recognize named entities in text.

1.3 Research Questions and goals

We firstly propose an overall goal for the work, which is:

Goal *Use pre-trained generative models for few-shot named entity recognition*

Our main interests are in the pre-trained and accessible generative models and the integration of visuals and language. How this affects named entity recognition, and especially the effects of adding visual data. We propose the research questions:

Research question 1 *Can prompting generative LLMs be a viable alternative approach for named entity recognition with little training data?*

$RQ1$ means having similar performance as current state-of-the-art methods under conditions where there is very little training data available.

Research question 2 *Is performance increased when integrating visuals for named entity recognition via pre-trained multimodal models?*

$RQ2$ asks if the addition of images can help make the models give better predictions. This would then be under the supposition that the images can benefit named entity recognition.

Research question 3 *Does more examples improve performance for the LLMs and multimodal models?*

$RQ3$ means that we are interested in if more examples, under the constraints that these models can handle, can produce better results or not.

1.4 Thesis Structure

The structure of the thesis is as follows. We stress that the background theory and related work are explained in brevity and are not supposed to be treated as an extensive dive into each topic but as a text adapted for the purpose of some foundational understanding for our experiment.

Chapter 1 - Introduction Presents the essential concepts, motivation, research questions, and goals.

Chapter 2 - Background Theory Presents essential foundational theory for the work. The relevant essentials of machine learning and deep learning, and the more specific theory for named entity recognition, visuals, language, and multimodal models which we will be working with.

Chapter 3 - Related work This chapter presents the related work concerning prompting generative models, and unimodal and multimodal named entity recognition.

Chapter 4 - Method and experiment This chapter lays out our approach. The methodology will enable us to execute a defined and concrete experiment and generate results.

Chapter 5 - Results Presentation of results from the experiment.

Chapter 6 -Discussion The results are discussed in light of the methodology and related work. Evaluation of the approaches, strengths, and weaknesses.

Chapter 7 - Conclusions and future work Conclusions are made concerning the work and research questions. Following, we present future work.

Chapter 2

Background Theory

2.1 Machine learning

This section briefly presents machine learning, its paradigms, and essential components of modern machine learning like artificial neural networks and deep learning.

2.1.1 Artificial intelligence and paradigms

Machine learning is, outside the obvious meaning of the word, the attempt to make computational systems adapt to new data, from which it then can do things like have better representational understanding, make predictions, and decisions, or in other ways leverage what it has learned. A broad topic, it can be described as a subfield of artificial intelligence. An essential component of machine learning is an attempt to make models leverage information directly instead of relying on explicit programmed instruction.

It is possible to describe Artificial intelligence paradigms as symbolic or subsymbolic. As a rule of thumb, the symbolic approaches use symbols with inherent representational properties to construct systems that can then learn. Subsymbolic then refers to systems that utilize values that do not inherently represent something, like a floating point number, to have what is called emergent properties.

The subsymbolic approach can be exemplified by artificial neural nets, where there are neurons that act as singular functions themselves, but with the interplay between other neurons, learning, and artificial intelligent expression can emerge. Still, while artificial neural networks can express behavior like answering questions, their explainability and logical mechanisms if they behave logically, are not plainly visible. The symbolic approach can be exemplified by an object-oriented programmed system that uses classes, and rules, and responds explainable and predictably.

The rapid development and use of new and better AI systems and models in recent years have largely been associated with the subsymbolic approach, leveraging new techniques, architectures, and larger models.

2.1.2 Types of learning

Meanwhile, there are not only different approaches to the construction of representation but also to learning.

Supervised learning refers to situations where we have labeled output data so that we can train the model for what it should output based on the training data. Perhaps the most intuitive form of learning is that it shows what is the input, and then what should be the output for a training example.
Unsupervised does not have labeled data but utilizes what's available to find and express patterns or features. This can, for example, be done by finding anomalies, clustering data points, or dimensionality reduction.
Self-supervised learning is quite similar to unsupervised learning in that it does not have labeled data, but self-supervised learning can be differentiated in that it, in layman's terms, uses techniques to make the input data trainable on itself. It, therefore, is unsupervised but also supervises itself. Techniques may include predicting masked words, contrastive learning, etc. Often useful because it can leverage unlabeled data to learn features or representations so that, for example, later fine-tuning on a labeled task will achieve much better performance.
Reinforcement learning works by not explicitly having input-output data but rather works through punishment or reward mechanisms.

2.1.3 Artificial neural networks

The foundations for artificial neural networks (ANNs) were laid by the Rosenbladt perception in the '50s. The perceptron takes inputs, weights the inputs, adds a bias and then is able to produce an output value depending on this configuration. It is the basis for ANNs and is essentially like a neuron in the neural network. The perceptron then works as a linear classifier.

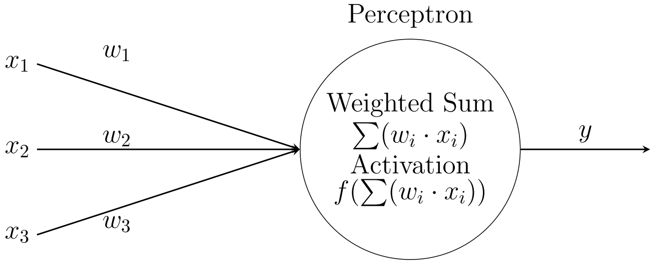

Figure 2.1: An illustration of the perception. The incoming inputs from the left get computed as a weighted sum before going through the activation function, producing an output.

The activation function maps the input to the output. This can often be a sigmoid function, like so:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.1}$$

Or it can be any number of functions. There are many, but some notables include the logistic function, Binary step function, Hyperbolic Tangent (Tanh), Rectified Linear Unit (ReLU), and the Leaky ReLU Parametric Leaky ReLU (PReLU) [60].

When we construct an artificial neural network, a fully connected one for example, we connect layers of neurons together to form successive layers which then can propagate values from the input layer to the output layer through the hidden layers.

Forward passes

The propagation of values from the input to the output takes place in what we call a forward pass. The forward pass for a single neuron can be mathematically represented as :

$$\boldsymbol{y} = f\left(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}\right) \tag{2.2}$$

where $\boldsymbol{W}$ is the weight matrix, $\boldsymbol{x}$ is the input vector, $\boldsymbol{b}$ is the bias vector, and f is the activation function [31].

In the network, the activation of a neuron i in layer l is given by:

$$a_i^l = \sigma\left(\sum_j w_{ij}^l a_j^{l-1} + b_i^l\right) \tag{2.3}$$

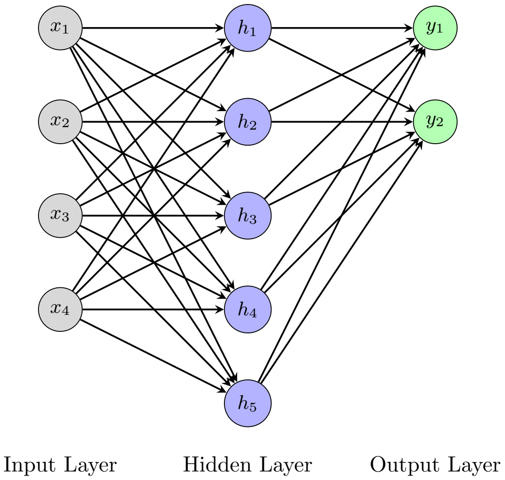

Figure 2.2: Shallow neural network illustrating the propagation of values from input to output.

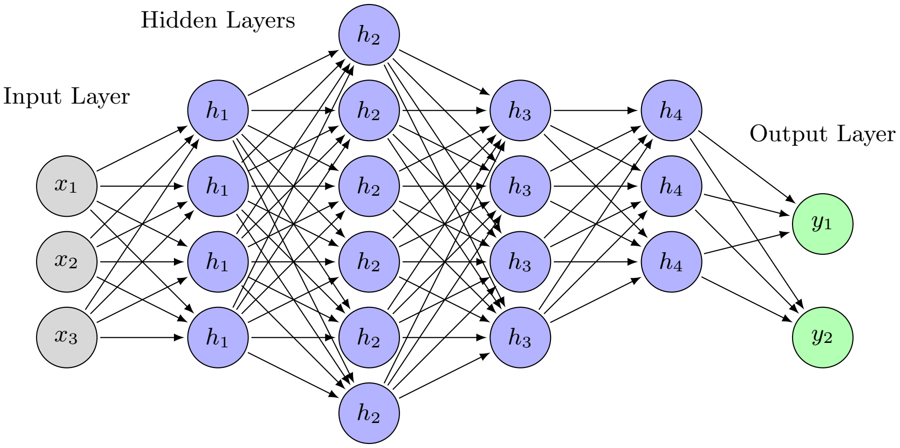

Figure 2.3: A bit deeper neural network with more hidden layers than Figure 2.2. The increased depth illustrates the added complexity.

where σ is the activation function, a_i^l is the activation of neuron i in layer l, w_{ij}^l are the weights between neuron i in layer l and neuron j in layer $l-1$, and b_i^l is the bias of neuron i in layer l [31].

2.1.4 Deep learning

When we then construct networks with multiple neurons in layers, and multiple layers which we want to make adapt to some data we start to approach what is called deep learning. To have the neural nets be learning from data, we need a mechanism for altering the weights depending on the error between the output we get, and the output we want. Here enters the loss function and the backpropagation algorithm.

Loss functions

Loss functions are a tool for how far off the network is from the mark. This can further be utilized for optimizing the network, by finding what we can change to minimize the loss function.

A loss function can, like the activation function, be set up in several ways. An example is using the mean squared error defined as [65]:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \hat{y}^{(i)})^2 \tag{2.4}$$

where N is the number of samples. $\hat{y}^{(i)}$ is the output of the system for the i-th sample. y is the actual output for the i-th sample [65].

Backpropagation

Introduced by Rumelhart et al. [73], the backpropagation algorithm can be used for training the network together with an optimization method. It works by first doing a forward pass, and then a backward pass which computes gradients throughout the network layers backward with respect to the loss function. With these gradients, optimization methods like gradient descent, stochastic gradient descent, or alike, make it possible to gradually by some factor α move the weights and biases of the network so that it should lower the loss.

The updating of the weights can be represented as:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha\frac{\partial E}{\partial w_{ij}} \tag{2.5}$$

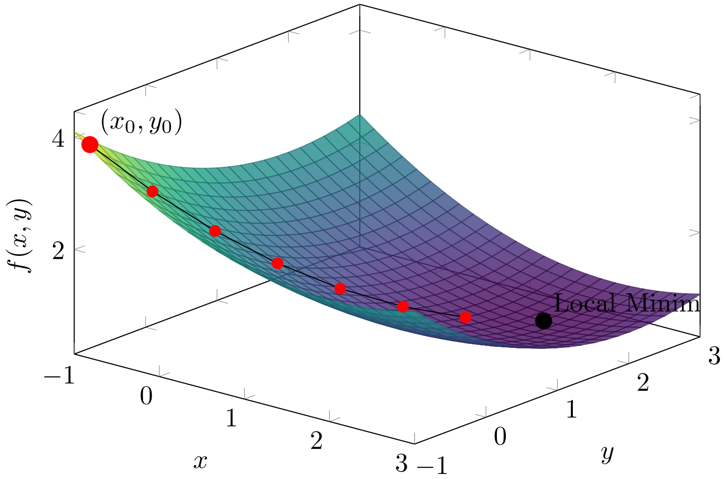

Figure 2.4: Incremental progress towards a minima using gradient descent on a 3D surface. We see the stepwise movement from origin towards the local minima.

where $w_{ij}(t)$ is the weight between neuron i and the neuron j at time t, α is the learning rate, and $\frac{\partial E}{\partial w_{ij}}$ is the partial derivative of the loss function for the weight w_{ij} [31].

Gradient descent

Gradient descent (SG) is an optimization method that can be described as updating parameters to find the minima of some function (loss function) [72]. By using backpropagation gradient descent enables the parameters to step closer and closer to local minima. There are varieties of SG (Stochastic gradient descent, Batch Gradient Descent) [71], but what we just explained is some of the fundamental essences of how deep learning is done, by iteratively moving towards an optimum based on tweaking parameters.

2.2 Vision language models

This section lays out the current techniques that are developed for handling visual and language modalities in themselves, and how they can be utilized together. Vision and language represent different modalities for data or information to be represented. In the case of language, we have the differentiation between

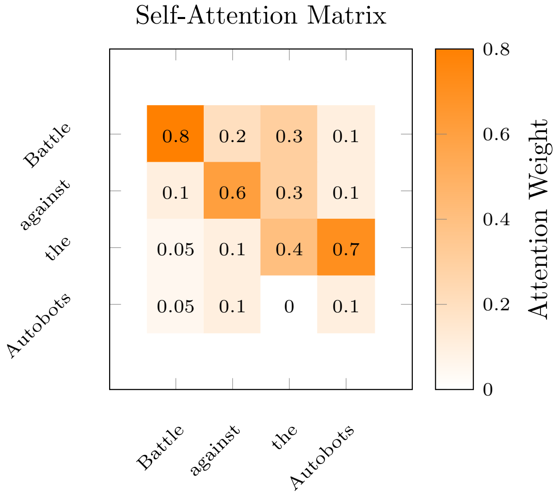

Figure 2.5: Visualization of Self-Attention across text input. Example values are fictitious.

written and oral, or textual and audio. With visuals, we have the presence or the absence of the temporal dimension, time, which differentiates images and video. For our work, we are concerned with images and textual information. This is a consequence mainly of named entity recognition normally being concerned with textual representation, as it is usually annotated word by word. Concerning the visual as images rather than video, it's a consequence of the available datasets and models at the current time. This does not mean that we think video for example is not a possible modality for future work (see section 7.2).

2.2.1 Unimodal techniques

We present the different foundational methods for handling visuals and language. We include what we believe are the essential modern architectures for visual and language by convolutions neural networks or transformers.

The self-attention mechanism

Attention as a concept is well known in fields like biology and psychology [41]. The human eye, for example, uses attention so that some parts of the visual field are more in focus [54]. The concept of attention has been incorporated into computer science, where we now have both attention and *self-attention* introduced by Vaswani et al. [82].

The self-attention mechanism is able to have each input element learn relations to all other input elements. It does this by using query Q, key K, and value V vectors constructed from inputs that then can do operations for each input in parallel before producing an output that is influenced by all inputs.

The self-attention mechanism means that we can score each element in the input relative to another for their respective pairwise attention scores:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.6}$$

where Q, K, and V are queries, keys and values, respectively, and d_k is the dimension of the key vector [82].

Transformers

The transformer architecture, introduced by Vaswani et al. [82] uses the self-attention mechanism to not only be able to understand the singular words in the sentence themselves but contextualize them. This means being able to infer from the surrounding future and previous words the meaning, which in isolation might be ambiguous or lacking context.

The architecture consists of an encoder and a decoder which are able to encode and decode textual representations, see 2.6. The benefit of Transformers over the previous popular methods, like RNNs [32] is that they can handle the entire input at once, meaning that they can infer the attention and contextualize a word using both previous and future words in parallel. The Transformer architecture proposed by Vaswani et al. [82] is laid out in Figure 2.6.

Vision transformers

Transformers have also been applied to images by Dosovitskiy et al. [23], but due to the computational complexity of having to do pairwise relations between all pixels in an image, the Vision Transformer sections the image into parts from where it can extract features.

There are many differences between Vision Transformers (ViT) and CNNs. But in general, the Vision Transformers are thought to require more training data to perform well than CNNs [20].

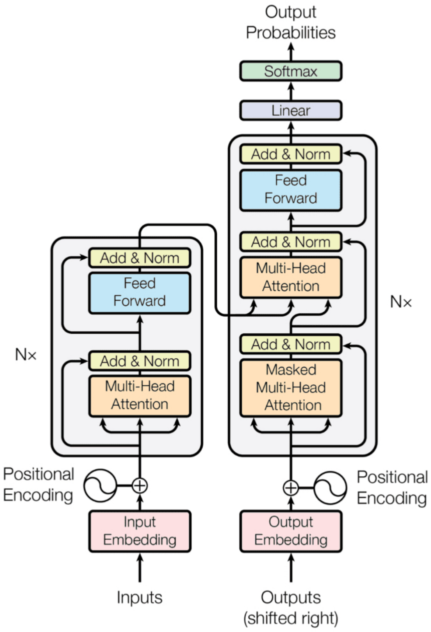

Figure 2.6: The Transformer model architecture by Vaswani et al. [83] ©CC BY-NC-SA 4.0. Encoder on the left, decoder on the right.

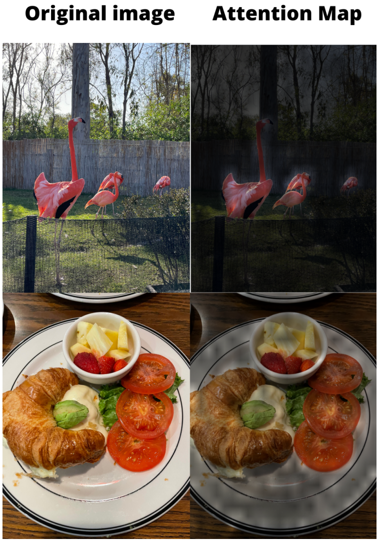

Figure 2.7: Illustration of vision transformer's attention map for different images, done using a ViT based on the work by [23].

Convolutional neural networks

Convolutional neural networks (CNNs) are, like neural nets themselves inspired by biology [62]. The neurons are clustered together, enabling the network to learn district features. This is especially useful for images since they often contain recurring shapes and forms. Further, CNNs are much more efficient than traditional ANNs when it comes to handling large image data [62].

The three main components of a CNN are the convolutional layers, the pooling layers, and finally the fully-connected layers [62]. The convolutional layer extracts features from the image, while the pooling layer then downsamples it. The fully connected layer then works as a classifier head, for example, classifying the image in a binary output shown in figure 2.8.

CNN's have long been the most popular choice of computer vision model, with milestones from the first backpropagation to train them by LeCun et al. [42] in 1989, to the advent of AlexNet by Krizhevsky et al. [40] in 2012 which demonstrated the effectiveness of CNNs compared to other approaches. AlexNet can be attributed to sparking off a series of deep learning models for vision [28].

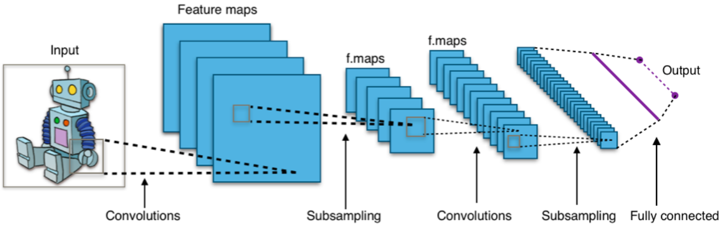

Figure 2.8: Illustration of the steps in a convolutional neural network classification task. Sourced from Aphex34 [3] ©CC BY-SA 4.0.

2.2.2 Multimodal integration

While vision and language models each separately have had extraordinary progress in the last decade, the merging of these modalities into integrated vision-language models has not been immediate. This can be attributed to the integrated models being very much dependent on the progress that happens in each modality and that the prospect of having generalizable visual-language models has not been seen as a very feasible prospect, at least not in the same way as it has become in the last years.

For an integrated model to leverage each modality it has to, in some way have an architecture that integrates them. Several approaches have been tried. Notably, PICA by Yang et al. [99] demonstrated that by captioning images directly to language models, they were able to obtain state-of-the-art results on visual question-answering tasks and perform well in a few-shot setting.

Fusion

Integrating modalities in end-to-end models may require fusion at the latent level. Fusing multi-modal information at some stage. This can be sectioned into early, mid, and late fusion [46]. Early fusion would mean that the modalities interact at the first stage/layer of the model, and so all the layers of the model would be multimodal layers.

Late fusion would mean having the different modalities go through the layers without interaction, meaning it would be similar to two separate models, but where the outputs of each model then are fused at the end somehow.

Mid-fusion would mean somewhere in between so that the model can interact with each modality separately at first, but at some stage, start having them interact together, and having some degree of learning at both a unimodal level and later also multimodal through the layers of the model.

Single stream then means having early fusion so that the end-to-end model handles the fused input end-to-end and learns the fused representation rather than individually [10].

Dual stream would then use two different streams for the two different data types and use some fusion mechanism, contamination, multiplication attention, or otherwise to leverage the two streams together at a later time [10].

Attention-Based fusion would then mean having an attention, co-attention, or transformer-like mechanism that learns the relationship between the modalities [46, 11].

This would then be somewhat similar to a transformer architecture binding the different modalities together in order to learn to weight the image features vs latent word embeddings against each other.

2.3 Pre-training and downstream learning

Downstream learning has become very popular, especially with the advent of Transformers, and pre-trained models leveraging self- or semi-supervised upstream training on large corpora have become a useful way to increase performance for later fine-tuning or few-shot learning [28]. In this section, we dive into these terms before we introduce few-shot learning and prompt engineering, which are important subjects for our work.

Transfer learning is leveraging learning from one domain to another. In this context want to leverage pre-trained models trained with large amounts of data to be able to say something specific about entities.

Upstream and downstream: For practical usage, we use the term upstream to refer to the self-supervised pre-training phase which trains the model and makes it learn features, but not with any specific task in mind. The term downstream then means training/fine-tuning the model so that it is able to solve the specific tasks we want leveraging the pre-trained features.

2.3.1 Techniques for pre-training

Here we highlight a couple of self/semi-supervised learning approaches that often lay the foundation for pre-training models.

Masked learning

Masked learning can be described as generating its own labels through masking part of the data so that the model can be trained to predict the masked data. An example of masked learning for NLP could be the sentence:

“I’m going to the [**MASK**] store to buy apples”.

We would then have the masked word **grocery** as a label, and the model would be trained to predict the masked token. When applied over large amounts of corpora, this technique has shown success, as demonstrated by Devlin et al. [21] with BERT, more on that in subsection 2.3.2.

Contrastive learning

Contrastive learning is, as the name implied, implemented leveraging contrast, or that is differences between two or more examples [12]. Techniques for contrastive learning are many since it only requires some augmentation of the data to create a contrastive example.

By augmenting data, it can create an augmented image that would function as a positive example, that the model should treat as more similar, and totally different images are treated as negative examples, which should be less similar.

2.3.2 Modern pre-trained models

Among modern pre-trained models (PTM) there have been some families of models or milestones that have played notable roles. We use these to showcase examples of modern PTMs and how they are applied.

BERT: A notable pre-trained model based on the pre-trained Transformer architecture is BERT, introduced by Devlin et al. [21]. BERT is composed of layers of encoders and trained using masked learning on large corpora, the English version of Wikipedia, and the Toronto BookCorpus [21]. This could then be leveraged for a wide range of tasks that benefited from language understanding.

ResNet’s Today, pre-trained convolutional neural nets like ResNet’s [30] have been popular vehicles in conjunction with fine-tuning for tasks like visual question answering, image captioning, etc [28].

GPT: The Generative Pre-trained Transformer from Radford and Narasimhan [66] uses the decoder from the transformer in addition to the encoder, so that it can both understand language input and generate new language. It showcased how self-supervised learning could be applied to the transformer architecture, with fine-tuning for specific tasks later. The GPT models were developed, and the successors as mentioned in 3.3.4 like GPT-3 [7] and ChatGPT demonstrate impressive ability in text generation.

2.3.3 Few shot learning

Few-shot learning means as the name implies, learning from only a few examples. The corresponding amount of "shots" and performance is compared with zero-shot, one-shot, and so on. The potential of few-shot learning is to be able to reach viable levels of performance on models for novel tasks, having only a few examples available [24].

But, this term is sometimes used for two different techniques. Few-shot learning by fine-tuning [27] and in-context learning [7] are two different variants that few-shot learning can imply. The former does traditional tuning of the weights but with a smaller amount of training data. In the latter, no weights are tuned. Still, the training data is provided as input at inference so that the models would leverage the additional information in the input to better handle the actual, without changing any weights or doing any tuning.

Fine tuning

Fine-tuning is more like traditional training (see chapter 2). The process can consist of leveraging a pre-trained model, unfreezing some of the last layers, usually some of the last ones, or all of them, and possibly adding an extra layer, a classification layer for example, to the model before training it so that the model can leverage earlier learned features but adapt to the new task [27]. The amount of few-shot training data that is used can vary greatly, with some attempting only a handful or several orders of magnitude more [47, 13].

Instruction tuning

Recently, instruction tuning has shown the ability to enhance large language model's zero and few shot ability [88]. This involves fine-tuning the models on datasets that contain instructions so that the models are more adept at adapting to natural language instructing prompts. Instruction learning has also been showcased to be effective for multimodal zero-shot learning by Xu et al. [95].

In-context learning

In-context learning has had a rising interest with the advent of especially large language models like GPT-3, introduced by Brown et al. [7]. Gao et al. has argued that pre-trained models can leverage in-context learning better than fine-tuning under some circumstances. On the other hand, it's also been shown that how, and how able large language models are to do in-context learning differs [90]. In-context learning has also been demonstrated to be effective in conjunction with instruction tuning [101].

Often for in-context learning the training data can mean only a handful or so examples as the number of examples that can actually be fed to the model is constrained by the maximum input length.

2.3.4 Prompt engineering

Prompt engineering is about designing and modifying prompts to models which can positively affect the probability of a getting a type of output at inference time [48, 29, 91]. Based on how the model is prompted, the output can change significantly depending on the sensitivity of the prompt, and the task can change the output and, for tasks, performance on them [4, 48].

Prompt engineering potentially makes models able to solve new tasks by having the task reformulated [79].

Prompt based learning

Because of the power the prompting can have on the ability of the models to perform well, in-context learning leveraging prompt engineering is sometimes called prompt-based learning [48].

The applicability of prompt-based learning is large, as shown by the numerous tasks exemplified in the survey by Liu et al. [48]. For prompt-based learning to be effective, there must be a design methodology for the prompting.

Prompt design

Prompting the models usually requires designing a prompt that will leverage the natural language capabilities the models have gained from being trained on a large corpus. [4] has found that QA-based prompt structures can be effective for prompting vs open-ended prompting in some cases. Meaning that when structuring prompts as:

Question: What does Parker work as?
Answer: Parker works as a clerk.

Can be more effective than an open-ended version:

Prompt: Parker works as a
Output: Clerk

Liu et al. [48] classifies prompt shapes into two categories, close, and prefix, whereby the former works by insertion into text, and prefix is completed at the end of the input. The prompting discussed in this work is exclusively prefix.

2.4 Named entity recognition

In this section, we explain the established methods for named entity recognition, including the current state-of-the-art approaches.

2.4.1 Flat and nested NER

The named entity recognition entity classes are usually either flat/exclusive, meaning that there can be no overlapping entity classes. Or, they can be nested so that an entity potentially can belong to several entity classes, where one class is part of another [25].

An example of such a hierarchically nested entity class is that an animal may belong to an animal class and a nested class of, say dogs inside the animal class.

It's also possible to construct entity classes that may overlap but are not nested inside one another [44]. But most approaches today are not very adaptable to this because they usually require one single entity classification for each token.

2.4.2 Granularity of entities

The granularity of the entity classes signifies the number of different entities we may use for a single NER classification. Many NER datasets, like CONLL [74, 96], leverage only four entity types, mentioned in section 2.4. Others may use more, but it is mentionable that the required labeling for a representative amount of entity labels in a dataset increases proportionally to the granularity of the dataset.

A survey by [96] also mentions that the data source can significantly impact the entity types that are present. Twitter datasets, by comparison, can therefore have more variable entity types [96].

2.4.3 NER today

Here we try to give some overview of how named entity recognition is typically approached in recent years.

Prevailing methods

The advancements of transformer-based models have also found their way to named entity recognition. Using pre-trained encoders, like BERT [21], and fine-tuning them for NER has been found effective [9], but more traditional methods like LSTMs and Conditional Random Fields are also used [9, 81].

Prevalent datasets

CONLL-2003 CONLL-2003 has four entity classes person, organization, location, and miscellaneous (other classes). It has become one of the standard datasets for named entity recognition [49, 2].

Evaluation

Evaluation is normally done using precision, recall, and F1-scores on the test dataset. Rarely accuracy is used as well [70]. Precision is defined as:

$$\begin{aligned} Precision &= \frac{T_P}{T_P + T_N} \\ Recall &= \frac{T_P}{T_P + F_N} \\ Accuracy &= \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \\ F1 &= \frac{2 * precision * recall}{precision + recall} \end{aligned} \tag{2.7}$$

T_P is true positives, T_N is true negatives, F_P is false negatives, F_N is false negatives [92, 97].

When done across the dataset as a whole the F1 score across all entities is called the micro F1 score, when done individually on each entity and then averaged so each entity counts equally independently of presence in the dataset, is called macro F1. For the rest of our work, we will use and refer to F1 as the micro F1 score, as the related research and ourselves use this as the main F1 score [81].

Chapter 3

Related work

In this chapter, we present the work that closely relates to our experiment. This includes few-shot learning for NER, multimodal named entity recognition, few-shot learning for large visual models, and multimodal tasks which we find somewhat close to ours.

3.1 Unimodal named entity recognition

Named entity recognition in a few-shot learning context has had recent advancements, so it's possible to reach higher-performant models in a low-resource setting [9]. In this section, we explore some of the related work on methods aligned with prompting and low-resource contexts.

3.1.1 Prompting for NER

Prompt-based methods have also been attempted for NER, notably by including either template for prompts to fill with training data [17], or manually designed prompts [86].

QA-NER

QANER, proposed by [47], provides a prompt-based approach to NER by prompting an extractive QA model for which entity/entities are in the text. Through converting the data to question-answering formats and leveraging a Bert model pre-trained on SQUAD2 [69] they showed significant performance, SOTA at the time for few-shot learning [47].

Templates for NER

Cui et al. [17] showed with TemplateNER that NER tasks can leverage a template format to be reconfigured as statements or prompts. This approach proved better than fine-tuning in their experiment.

LightNER

Chen et al. [13] presented a method where they attempt to make domain transfer from one NER dataset to another more efficient by using the generation of entity span together with entity types.

3.2 Multimodal named entity recognition

Moon et al. [57] and Zhang et al. [105] introduced the first public datasets and perhaps the first notable examples of leveraging images for named entity recognition successfully. Further work [53, 103, 86, 102, 76, 104, 78, 11] has showed that with the publicly available datasets, the images can enhance performance. Showcasing that images can help with named entity recognition on the corpus by providing additional information. This is also very much in line with the thinking that leveraging multiple modalities, in general, can provide more information that can be useful for tasks at hand [59]. On the other hand, this is not always the case as additional information can provide noise, the dataset could have images that actually conflict with the task. There could for example be a dataset that shows the CEO (person) as an image when describing the organization textually, which then could sway the classification towards a person, while really being an organization. This correlation between images is discussed by Chen et al. [11], where they constructed a dataset from Snapchat data hypothesizing Snapchats might have more alignment between images and text than the publicly available Twitter datasets.

3.2.1 PromptMNER

PromptMNER [86] is the first prompt approach to MNER which tries to leverage a prompt-based technique on NER with images. It follows from the work done by Liu et al. [48]. They provided general prompts corresponding to entity types used to score image vs prompt co-similarity. Thereby extracting relevant information from the images. The similarity score between the prompt and image was used to guide the language model for classification. PromptMNER achieved state-of-the-art results on the Twitter2015 and Twitter2017 datasets [86].

3.2.2 Other approaches

Several other approaches than prompt-related ones have been tried for MNER. A commonality is that they use fine-tuning on the Twitter datasets, and utilize quite specialized techniques targeting MNER. As such, the works' relevancy seems limited, and we will therefore describe only a few very briefly.

Wang et al. [84] extracted text from images so that visual clues could be represented in language through captioning and optical character recognition, and added to the textual information. Sun et al. [77] uses forget gates to select visually relevant information. Zhang et al. [103] uses graph-based techniques for multimodal alignment. Chen et al. [14] uses a dynamic gating strategy that enables them to use visual information as guidance for textual processing. Xu et al. [94] design cross-modal-alignment and matching modules to handle the two modalities. And Zhao et al. [107] uses graph convolutional networks on each modality before fusion.

3.3 Prompting large models

As more generalizable and larger generative models become available, utilizing them for a larger range of tasks has become more achievable. This section describes relevant work that uses these types of large unimodal or multimodal models.

Prompt engineering has had advancements as these models have become larger since there appears to be a correlation between the size of the model and the ability to handle in-context learning [7].

3.3.1 Chain-of-thought (CoT) prompting

Chain-of-thought (CoT) prompting involves adding intermediate steps of reasoning that have shown the ability to produce better outputs from models by making them have to elaborate reasoning behind outputs [89].

This can, for example, include steps in a calculation of numbers or an equation, such that the model is able of handling more complex tasks. Generalizable models have shown great ability to produce sentences and text by putting one word in front of another, but for reasoning tasks like arithmetic scaling up the model has not always had the same performance improvement [89]. Seemingly, the way generative models work today produces some output that looks "something like" what is expected but has no logical anchoring in its reasoning, such that it is fully possible for the model to output 2+2=5, because it produces similar outputs to what is in the corpus, but does not actually have mathematical intuitions. By adding intermediary steps, these foundational intuitions, such as 1+1=2 can aid more complex prompts.

3.3.2 Prompt tuning

Recently, prompt tuning has emerged as an alternative to prompt engineering and fine-tuning [43, 50]. While prompt engineering essentially seeks to find the best prompt through the methodology and manual experiment, prompt tuning rather engineers this by having an additional encoder being tuned to output encoded representations of prompts that fit the prompted LLM the best [50].

3.3.3 Flamingo

Flamingo is a model by DeepMind, introduced by [1]. It showcases the ability of the visual-language model to do in-context few-shot learning, as demonstrated in the figure Figure 3.1. This was a decidedly important step in showcasing visual-language few-shot capabilities. Further, most openly accessible generalizable visual-language models mention Flamingo and usually benchmark against it [45, 39].

Flamingo fuses pre-trained and a frozen language model with a visual model by the inclusion of Perceiver Resamplers for the visual modality outputting visual tokens which then the gated cross attention layers fuse with the language modality [1]. These components are then trained to align the frozen unimodal language and visual models.

By leveraging an NFNet-F6 vision encoder and Chinchilla language models Flamingo was able to perform near or SOTA at tasks, using only few-shot or zero-shot learning [1].

3.3.4 Latest GPT models

Large language models, with special focus on GPT models, and notably GPT-3 has shown an ability to benefit from prompt engineering [7]. GPT-3 was explicitly published as a few-shot learning capable model [7], and has since become an important model demonstrating the advancement of large language models [36].

Since the recent popularity of chatGPT, prompting GPT models has become even more accessible. After 2 months there were approximately 100M users with

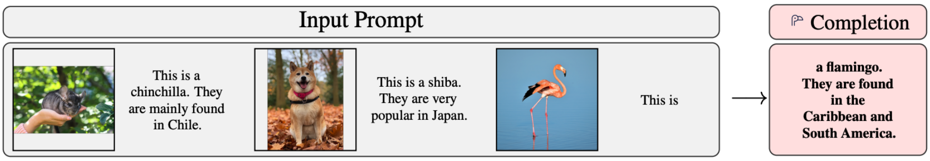
Input Prompt
This is a chinchilla. They are mainly found in Chile.
This is a shiba. They are very popular in Japan.
This is
Completion
a flamingo. They are found in the Caribbean and South America.

Figure 3.1: Demonstration of Flamingo-80B. The figure shows how Flamingo was able to adapt to new inputs and contexts and produce useful outputs with very few examples. Sourced from Alayrac et al. [1].

access to ChatGPT [56], interacting with the model through prompting.

Since then the GPT-4 model has also been released, which is able to handle images in addition to text input [61]. We infer from this that the development and popularity of leveraging large language and visual-language models have been on the rise and that with more accessibility, the experimentation with prompting such models may continue [51, 8].

3.4 Similar applications of MLLMs and LLMs

Our approach builds on the advent of generalized visual-language models, but there are also specialized models and tasks which by virtue of their similar inputs and outputs are presented here.

3.4.1 Document Image Understanding (DIU)

Document Image Understanding seeks to handle textual content with images of documents for information extraction. Models from Microsoft like LayoutLM have been adapted to question answering, meaning it can answer questions using textual context and images, extracting the answer from parsed text from the image. The key differentiator for us is that the model is trained on document images, and extracts text from them instead of leveraging the context.

Still, if there was a dataset where the textual content was classified as in an image form, these models could be used. Work by [87] used such models to do few-shot entity recognition. But again, the images are documents and are not used as visual cues in addition to the actual text.

3.4.2 Visual question answering (VQA)

Visual questioning involves being asked questions about images, and for the model to answer them. Some VQA models also incorporate a textual context input. So that these models in theory can be prompted with a question, a text, and an image. But for this task, the models are trained so that the image is the main focus. Meaning that the questions are concerned with the image, and not the context, the context only guides the VQA.

3.4.3 LLMs with image captioning for VQA

Image captioning models produce textual descriptions of images. These have been used in conjunction with large language models in the work done by [99] where they present the PiCa method. They showed that using captioning and then feeding it to an LLM like GPT-3 could be an effective way to do VQA.

This is an interesting and flexible example of using separate visual and language models in conjunction so that the LLM can leverage the visual modality.

Chapter 4

Method and experiment

In this chapter, we present our method and our experiment which will be a concrete implementation of the method. We describe our method for how we adapt NER to generative question-answering. From there we describe the specific datasets, models, prompts, and other factors for our concrete experiment. These other factors include but are not limited to preprocessing, inference, postprocessing, and evaluation.

4.1 Adapting NER for open-ended prompting

Since we are approaching this task from a relatively novel angle, using open-ended question answering, we are dedicating a section for describing how we aim to make the datasets and the task amenable to open-ended and generative answering.

4.1.1 Templates and instructing prompts

For our use case, we construct differing template prompts. Template-based prompts use random examples from the datasets. In addition to the prompting template the prompts for generative models are fed some instructions in prompting.

Instructions

The prompt to be answered has appended structure when fed to a generative model, as follows:

PROMPT	**TEXT_N**	**ANSWER_N**
PROMPT[person]	TEXT_N	*ANSWER[person]_N*
PROMPT[organization]	TEXT_N	*ANSWER[organization]_N*
PROMPT[location]	TEXT_N	*ANSWER[location]_N*
PROMPT[miscellaneous]	TEXT_N	*ANSWER[miscellaneous]_N*

Table 4.1: Example of asking for each entity type. For the CONLL format [74] one input would have the four following prompts.

Intention: Explaining that we want to do named entity recognition.
Image specifier: When using an image the prompt has an added sentence specifying that the image is supposed to aid the question. When applicable an image token is also inserted.
Negative output establishment: The prompt specifies what to output if there is no answer or no entity.
Multiple answer clarification: A sentence specifying how it should answer if there are multiple entities.
Entity type enumeration: A sentence enumerating the entity types in the dataset.

This additional information is also suspect to the way its framed in natural language. We have opted for some examples we believe to be concise and clear and resort to discussing further considerations regarding the prompt engineering in 6.

4.1.2 Entity classes and N-prompts

The entities are converted to their natural language equivalents as in work done by [47]. PER becomes ”person”, etc. In addition, we do, also as [47] and prompt a single input N times when there are N entity classes, each time asking if there is a certain entity class in the input. This that for N entity classes we run inference N times across the dataset.

4.1.3 Handling answers

The usual structure of NER prediction and evaluation was covered in section 2.4. We understand that the open-ended format makes it necessary to handle the answers differently.

Interpreting answers

The specifying prompt structure described above 4.1.1 exists so that the model hopefully adapts to our choices of format. For example, when we ask it to answer if there are any person entities it then answers ”none”. Or that if there are multiple, it answers ”A” & ”B”, specified to deliver multiple answers with ”” between them.

The inputs and predictions for the tasks are therefore both structurally different from a classifier. How we handle that we are dealing with an open-ended model is the following:

1. We post-process the answer, adapted to how the model answers the question, stripping away text it consistently outputs indifferent to what it's being fed. This differs from model to model, but can, for example, be appending a newline token to its answer.

2. When running the evaluation, if the answer from the model is not to be found in the input text, we discard the answer and considered it the same as answering negatively, *“none”* for example. We, therefore, **look** in the input for a match. This restricts the model to the input text. We stress that we consider the answer by omission, so that if the model does not answer (so if the output is empty) we considered it a negative answer. Also when there should be multiple predictions. So that if the ground truth has entity A and entity B, and the model answers just A, we consider it to have a true positive on A and false negative on B.

3. We are not getting placement specifying output like a classifier or closed-QA model would. When the model says *Apple* we don't know if it means *Apple* at index **X**, and there could be two mentions of “*Apple*”, that just one, both, or neither could be entities. In this case, we treat the first outputted answer as corresponding to the first match in the input. We then require multiple outputs when there are repeating examples of the same entity in the input. We exemplify this below:

   “*Apples* cool new iPhone is the best, also I like *Apples*”.

   In the first case of *Apples* we are looking at a slight misspelling that nonetheless is supposed to be treated as an entity, while the latter occurrence should not. If the *Apples* were reversed, and the latter was an entity, we could not give it a correct entity classification without further methods. This edge case relies on the input having an entity and a non-entity spelled exactly the same, including the same capitalization. This is further discussed in chapter 6. We are nothing that this edge case does not occur anywhere in our datasets.

Adapting to flat NER

For flat NER, there can be no overlapping entity classes. To achieve this, we think of three methods that are believed to be applicable.

1. Get the scores for the generated outputs and for conflicts pick the one with the highest score.
2. Running an additional prompt for disambiguation. So that if there are any overlapping entity classifications, the model is essentially prompted to choose.
3. Automatic handling through preference or negation. So that either an entity class would take preference, for example, if MISC always has the lowest preference. Or/and that if some entity classes overlap, we throw them out.

For our experiment, we resort to the first. This is because we want a leaner processing pipeline and not add more additional intermediary steps as required by 2., and we believe that using 3. will lower the performance of the models because it will throw out correct classifications relative to 1.

In nested NER, classes can overlap, and would therefore rather involve prioritizing the nested entity. In that way, our approach is also considered applicable to nested NER, or NER with discrete overlapping entities.

4.1.4 Examples for few-shot learning

For our in-context few-shot learning, we need to have some valid examples which the models can take inspiration from and apply to the inputs. Here we present how we do this given the visual and language modality.

The visual modality in examples

For our experiment, we are restricting the in-context examples to using only the language modality. This is because some models can only handle one image, and feeding concatenated example images with the input image was seen as adding a lot of complexity to the interaction between the visual and language modalities if the model in addition was to be instructed to differentiate and learn from them. Nonetheless, some of the models, like FROMAGe and OpenFlamingo are able to handle interleaved images and text, and the Flamingo model [1] also showcases

Question + Text	Answer
person? + Text1	person-entity
location? +Text2	location-entity
organization? + Text3	organization-entity
miscellaneous? + Text4	miscellaneous-entity

Table 4.2: Example of a 4-few-shot/1-set training set in our experiment for when there are 4 entity types.

this ability to have multimodal training examples. Meaning it's possible in section 7.2.

Rather we focus on instructing the models to utilize the image for the input to be answered, and for it to leverage its pre-trained abilities for visual feature extraction to make it useful for entity disambiguation. Considering our dataset, it's not clear that our data leverages the images explicitly, as mentioned by [57] the Twitter datasets. The usage of example images was therefore considered out of scope for our work and is discussed in section 7.2.

Textual examples

The few shot examples are inserted before the input prompt. They are structured the same as the prompt, with the addition of a completed answer.

Normally, few-shot NER would consist of having 1-shot be a collection of examples so that all entities are represented 1 time, this would mean that for N entities there could be N examples in 1-shot. For example in the work by [47] where an M-shot example means one of each entity class, and also possibly adding negative examples. Because we are experimenting with in-context learning the maximum of possible examples is much smaller than for fine-tuning because of the constraints on input size, and we therefore will look more granularly at the N-shot examples.

In our work, we, therefore, adopt 1-shot as 1 singular example from the training set, where for N-entities, N-shot would therefore signify the same meaning as 1-shot in much other work. This is explicitly shown in the results as well. Exemplified below in table 4.2, where each text from 1-4 has a corresponding positive answer in the form of a matching entity class. This is to include the full range of examples which any random example input can encounter.

English	CONLL-2003		
	Training set	Development set	Test set
Articles	946	216	231
Sentences	14,987	3,466	3,684
Tokens	203,621	51,362	46,435
LOC	7140	1837	1668
MISC	3438	922	702
ORG	6321	1341	1661
PER	6600	1842	1617
Total entities	23499	5942	5648

Table 4.3: Description of CONLL-2003 dataset. The used dataset is the English variant [74].

4.2 Datasets

In this section we describe the datasets we utilize, we first introduce unimodal language datasets and multimodal ones which include images.

4.2.1 Unimodal language datasets

To compare our methods without visual to existing few-shot methods in literature [47] we use the CONLL, MIT Restaurant, and MIT Movies datasets as benchmarks. Statistics for the CONLL dataset are in Table 4.3, MIT Restaurant in Table 4.4 and for MIT Movies in Table 4.5:

4.2.2 Multimodal datasets

The availability of openly accessible MNER datasets is at the current time quite limited. We find two openly accessible datasets, Twitter2015 by Zhang et al. [105] and Twitter2017 by Lu et al. [53]. We note that another dataset was created by Moon et al. [57] at Snapchat, but that this is not openly accessible. Common for all of them is that they use the same four entity types from CONLL-2003. We discuss the limited amount of datasets in section 7.2.

Twitter-15 and Twitter-17

Twitter-15, published by [105] is a public multimodal dataset based on tweets. Twitter-2017 by [53] is also a public dataset based on tweets. These are the two

English	MIT Restaurant		
	Training set	Validation set	Test set
Sentences	6900	760	1521
Tokens	63269	7256	14256
Location	3355	462	812
Cuisine	2532	307	532
Amenity	2249	292	533
Restaurant_Name	1755	146	402
Dish	1353	122	288
Rating	987	83	201
Hours	871	119	212
Price	655	75	171
Total entities	13757	1606	3151

Table 4.4: Description of MIT Restaurant dataset, the training and validation set is the split of the original training set.

English	MIT Movies		
	Training set	Validation set	Test set
Sentences	6816	1000	1953
Tokens	138462	20361	39035
Plot	5666	802	1577
Actor	4419	591	1274
Genre	2997	387	789
Year	2391	311	661
Director	1581	206	425
Character Name	921	104	283
Opinion	723	87	195
Origin	669	110	190
Relationship	511	69	171
Award	268	41	66
Quote	113	13	47
Soundtrack	45	5	8
Total entities	68393	2726	5686

Table 4.5: Description of MIT Movies dataset. The training and validation set is the split of the original training set.

Entity Type	Twitter-2015			Twitter-2017		
	Train	Dev	Test	Train	Dev	Test
Person	2217	552	1816	2943	626	621
Location	2091	522	1697	731	173	178
Organization	928	247	839	1674	375	395
Miscellaneous	940	225	726	701	150	157
Total	6176	1546	5078	6049	1324	1351
Number of Tweets	4000	1000	3257	3373	723	723

Table 4.6: Twitter-2015 and Twitter-2017 datasets

publicly available MNER benchmark datasets. The test sets are used for benchmarking; the metrics for the datasets are in Table 4.6

As we can see, and as we will see in all the available datasets, there is an unequal distribution among entities. This disparity is something we take special notice of and that we discuss further in chapter 6.

4.3 Models

The multimodal models leverage existing language, and visual models, bridging the modalities with some alignment procedure. The models have in common that the available pre-trained checkpoints all use a limited set of large language models, and they all use CLIP [67] as a vision model. We, therefore, elaborate on the language models and CLIP before we describe the multimodal models. This is useful because we also use the unimodal language models in themselves in experiments for comparison with the multimodal models. The repositories used, and the origin of weights for the pre-trained models can be found in section B in the appendix.

4.3.1 Unimodal models

The generative large language models used are Flan-T5, OPT and LLaMA with sizes ranging from 2.7B-11B parameters. The visual model for all of the multimodal models is CLIP. The unimodal language models are also tested on their own, in addition to the multimodal models that use them. In addition, we reproduce the work done by Liu et al. [47], for a comparative baseline using a BERT model trained on the SQUAD2 dataset.

Unimodal Language Models	**Parameter size (B for billions, M for millions)**
Flan-T5 XL	3B
Flan-T5 XXL	11B
OPT	2.7B
OPT	6.7B
LLaMA	7B
BERT-large	336M

Table 4.7: Large language models used and parameter sizing.

Comparative baseline model

By using a BERT-based model pre-trained on SQUAD2, we reproduce the experiment by Liu et al. [47], whereby we can compare our methods directly to theirs using our prompts and training examples. Because their implementation only used training sets of N-entity (so sets of all entity types, not singular examples), we do that in our experiment as well for this one model. Notably, this model will be fine-tuned, and not use in-context learning, meaning that the comparison is for the method with respect to another available method with the same low-resource data. We used the GitHub repository by Paiheng [63] as inspiration for our implementation.

Flan-T5

Flan-T5 was proposed by Chung et al. [16] as an improvement to the existing T5 models [68]. It is instruction fine-tuned. This means tuning the models on instructive data which results in the models being able to better adapt to new zero or few-shot tasks. The instructive data contains both zero-shot instructions, few-shot examples, and chain-of-thought prompting [16].

The Flan-T5 models are therefore pre-trained on both large amounts of unlabeled data, like T5, and also on a large collection of instructive datasets, Chung et al. [16] mention SQUAD as an example.

OPT

The Open Pre-trained Transformers (OPT) models were introduced by Zhang et al. [106] from Meta AI with sizes ranging from 125M to 175B parameters. The models were pre-trained on a large volume of mostly English text. They are open-sourced and meant to contend with GPT-3. The intention is therefore to

Multimodal Model	Language Model	LM-Parameters	Vision Model
BLIP2	Flan-T5 XL	3B	CLIP
	Flan-T5 XXL	11B	CLIP
	OPT	2.7B	CLIP
	OPT	6.7B	CLIP
FROMAGe	OPT	6.7B	CLIP
OpenFlamingo	LLaMA	7B	CLIP
mPLUG-Owl	LLaMA	7B	CLIP

Table 4.8: Multimodal language models and their corresponding LLMs and CLIP visual model. We note that all models were released this year (2023).

make publicly available LLMs for research. Zhang et al. [106] report comparative performance with the same amount of parameters, but varying on the task.

LLaMA

LLaMA is also an LLM made public by Meta AI, introduced in the paper ”LLaMA: Open and Efficient Foundation Language Models” by Touvron et al. [80]. Trained exclusively on public data, the authors notably report that the LLaMA models can achieve comparable performance on many NLP benchmarks as larger models like GPT-3 and PaLM. LLaMA weights are licensed, and we received licensing for this work.

4.3.2 Multimodal models

The multimodal models used are BLIP2, FROMAGe, OpenFlamingo, and mPLUG-Owl. A notable factor is that publicly available large multimodal models like these, which are able to flexibly generate textual input with both language and visual input are a quite recent phenomenon and that all of the models mentioned here were released this year. Notably, they all leverage distinct large language models and vision models and then integrate them later, meaning they use something akin to a dual-stream approach. In Table 4.8 we show an overview of the models where parameter sizing corresponds to the language model size.

CLIP

CLIP (Contrastive Language-Image Pre-training) is used in all the multimodal models as the model handles the visual component. CLIP is one of the OpenAI models introduced by Radford et al. [67]. It has been compared as a visual model to the language models GPT2 and GPT-3 in its ability to do zero-shot predictions [67]. Our CLIP version leverages a ViT (Vision Transformer), an image encoder, and a transformer text encoder, which can align similarity scores for vision and language.

BLIP2

BLIP2 (Bootstrapping Language-Image Pre-training) results from work done by Li et al. [45] where they freeze pre-trained language and vision models, bridging the modalities with a Querying Transformer (Q-Former). The Q-Former is a transformer trained to align the visual modality to language. It does this by inserting querying vectors and training the Q-Former to be best able to pass on information from the vision encoder to the LLM that will improve the output (of the LLM).

BLIP2 is reported to have zero-shot image-to-text generation capabilities, meaning that when combined with a language model with zero and few-shot learning capabilities, it can be useful for multimodal zero and few-shot learning [45]. BLIP2 can leverage a variety of "off-the-shelf" language and language models. We have chosen to leverage the available Flan-t5-XL, XXL and the OPT-2.7b and OPT-6.7B models for language. CLIP vision model.

FROMAGe

The Frozen Retrieval Over Multimodal Data for Autoregressive Generation (FROMAGe) model utilizes image captioning and contrastive learning to visually ground the language model [39]. They do this by keeping the language and visual models frozen and training linear layers to map from one modality to another.

FROMAGe has been reported as having strong few-shot and in-context capabilities [39]. We leverage the available pre-trained model, which uses an OPT-6.7B language model and a CLIP vision model.

OpenFlamingo

The OpenFlamingo model [5, 108] is, as the name implies, an open-sourced implementation of the Flamingo[1] model. It bridges the modality like the Flamingo model, utilizing Perceiver resamplers for handling visuals and outputting visual tokens and cross-attention layers. The cross-attention layers are inserted and interleaved with the blocks of the LLM and trained to handle the input from both vision and language. The mechanism is then an attention-based approach. The cross-attention layers were constructed such that at initialization the vision inputs would not interfere with the LLM and therefore, when merging the modalities the vision modality did not interfere with the language model before being trained to do so supposedly improving performance [1].

The group behind OpenFlamingo Awadalla and Gao [5] describes the goal of the model development to match the power of GPT-4 [61]. The model can handle interleaved images and text, and they describe it as having in-context few-shot learning capabilities [5].

We are working with an early released pre-trained model, the OpenFlamingo-9B, which leverages a LLaMA 7B model and a CLIP vision model, sourced from the OpenFlamingo repository Awadalla et al. [6].

mPLUG-Owl

mPLUG-Owl developed by Ye et al. [100], leverages a modular technique for language and visual models. They use a two-stage method for processing the visual input before feeding it to the LLM. The modality-merging approach is therefore concatenation based. We use the weights from the instruction-tuned combined LLaMA LLM and a CLIP visual model from their huggingface[93] checkpoint(see appendix B).

4.4 Experimental setup

This section describes the technical setup and process of our work. It explains the data pre-processing, prompting, in-context learning, model configuration, pipeline setup and evaluation metrics.

4.4.1 Pre-processing

We don't do any significant alterations to the corpus, as we are not looking to alter the data integrity. But we do construct new prompts which fit our new format. Entities from the text are extracted from examples, then linked to their corresponding entity. These then become answers for the text when that entity is asked for. We create examples for each of the N-entity types, so that for each entity there is a range of examples.

No.	Prompt
1	What word(s) in the text corresponds to a ENT entity?
2	ENT in text?
3	ENT entity in text?
4	What is the ENT entity in the text?
5	What is the ENT in the text?

Table 4.9: Question prompts to identify $< ENT >$ entities in text.

Further, we process the given prompting structure for each 0-8 shot example case so that there are fully constructed prompts in the data frame, with an associated image path.

4.4.2 Prompt generation

There are multiple snippets of text involved in our prompt. The templates we use for our prompts are as follows:
$[Training - examples] + Instruction : \{\} + Question : \{\} + Text : \{\} + Answer : \{\}$. Where training examples denote examples in the form Question: {} Text: {} Answer: {}.

Question prompt templates

For our question template prompts, we take inspiration from some of the templates by Liu et al. [47], Cui et al. [17] and add our own (in the case of prompt no. 1) using more detailed language. The question templates are shown in Table 4.9.

Instructions

Most obviously prompts that can make or break the performance of the experiment, we did piloting experiments to better decide on the instructing text.

As mentioned in methods there are some instructions we think of, initially, we declare the intention of the prompt, provide negative output establishment, to instruct it what to output if there is no answer (it would degrade performance to get outputs matching the input), multiple answer instruction, how to answer when there are multiple right answers. And in the case of visuals being present what the image is for.

Instruction	**Prompt**
Intention	We want to do named entity recognition.
Negative output	If there is no entity the answer is none.
Multiple output ’ between.	If there are multiple answers, output them with ’
Image specifier	The image is meant to help with the question.
Entity types	NA

Table 4.10: Instructions in the prompts to guide the prompting. Instructions come before the question and after the training examples.

After some piloting experiments and with conciseness and clarity in mind found the following to be effective instructions:

Interestingly, we found that performance as a whole actually degraded if the input contained the enumeration of all the entity types. So the entity that is being asked about is really what seems to matter.

Training examples

We construct training examples by using randomly selected examples from the training data and using our prompt templates. The examples are created at random from the dataset, but a new entity class is added each time so that the training examples never contain more than a 1-more in difference regarding the number of entity types. We do 5 random samplings of prompts for each type of few-shot example creation to minimize the risk of bad sampling [47].

For experiments, we are limited by the length of the input, which is possible or practically useful for the models to take in. The part of the prompt that eventually takes up the most tokens is the examples at eight-shot length. Some of the models have input size constraints at 512, which are not necessarily hard constraints as they can be altered, but that reflects the size of their training. By sampling some generated examples, we found that eight examples in addition to the relevant prompt would be, on average, at around 512 token lengths. After this, the performance of some models would drop due to not being trained to handle such long inputs.

4.4.3 Hardware and running inference

We run our pipelines on A100 GPUs with either 40 or 80GB of VRAM, depending on the need. OpenFlaming quickly becomes too large when fed with longer inputs and ran only on 80GB VRAM.

Batching was done at varying degrees during development. For some models, when fed with longer inputs from the examples, it is not viable to do any significant batching. On the other hand, at zero-shot, some models could do upwards of batches of 64. For the final experimental results, we used bathing of 1 to avoid any suspicion of unstable results, which occurred for some models when batching. We refer to Table A1 and Table C4 in the appendix for more details on inference parameters and notable software used.

4.4.4 Evaluation metrics

The standard evaluation method for named entity recognition is the stringent metric, as laid out in section 2.4. Other metrics include a more nuanced approach, giving credence to partly-correct predictions.

Stringent NER metric The stringent metric only considers a prediction that exactly matches all tokens and the right entity type as correct. This means that for a sentence like:

Maria Carey is looking for her next tour destination.
We would only consider the prediction correct if it predicts Maria Carey, not Maria, and not Carey.

The standard CONLL [74] dataset usually only considers the stringent NER metric, and because of the large number of data points in the result we, for this work, only consider the stringent metric.

Chapter 5

Results

In this chapter, we present the results of the experiment according to the description and methods from the previous chapter 4. This chapter is sectioned by presenting the baseline results first, then experiments with our models for datasets containing only language, and the datasets containing language with the addition of images. Lastly, we present results of particular interest in more detail, with more details for some results appended in section D in the appendix to reduce the already numerous score results that are presented. All F1 scores are micro F1 scores, if not specified otherwise.

The following tables contain as mentioned micro F1 scores, these are the mean of the 5 runs where the training examples are randomly sampled (but the same random samples across different models and prompts). This is also where the deviations from the mean come from, as the inferential parameters make sure the models perform the same under reproduction. As a consequence, the zero-shot experiments result in no variance/deviation. As we have a large number of F1 scores, we saw that the significant results for our findings in the discussion needed only zero-shot results, while the few-shot results rather would be demonstrative, and congruent with the zero-shot results. As a consequence, for deviations, we report from all experiments that **all zero-shot results in our experiment has ±0 deviation from the mean**. Meaning we can more easily and statistically significantly compare the zero-shot results against each other, and this is sufficient for our discussions and conclusions. Still, we append the deviations for the unimodal dataset experiments in appendix section E

The prompting templates used are the ones earlier mentioned in Table 4.9, and

Dataset	Micro F1 with deviations ±	Configuration
CONLL	0.56 ±0.01	Blip2-Flan-t5-xxl, 3-shot with prompt no. 2
MIT Restaurant	0.45 ±0.03	Flan-t5-xxl, 1-shot with prompt no. 2
MIT Movies	0.59 ±0.01	Blip2-Flan-t5-xl, 8-shot with prompt no. 5
Twitter2015	0.64 ±0.00	Blip2-Flan-t5-xl using visuals, zero-shot with prompt no. 4
Twitter2017	0.60 ±0.00	Flan-t5-xl not using visuals, zero-shot with prompt no. 4. Also, Blip2-Flan-t5-xxl not using visuals, zero-shot with prompt no. 2.

Table 5.1: Short summary of the best performing configurations for each dataset.

the numbering of prompts in the results corresponds to the number given in that prompt template table. Unless specified all the results also use the instructions from Table 4.10.

More technical details for the experiment can be found in the appendix section A. For brevity with regards to the best scores, we include Table 5.1 which showcases the best scoring configurations in our experiment.

5.1 Baseline results

As our QA-NER baseline handles a few shot sets and not singular examples, we provide all the baseline results for the language datasets (CONLL, MIT Restaurant, and MIT Movie) first in this section. The results are reported in Table 5.2. For these results, only sets/shots, which means N times the entity type are used, where N is the number of entity types. This is the same method as in the original implementation, as opposed to our subsequent results for generative models where we use 0-8 singular examples, as a consequence of restrictions on input length for in-context learning versus fine-tuning which is adopted for QA-NER.

As the results show, maybe unexpectedly, the micro F1 scores often go down with 1 or 2 sets of training examples. We validated that our method gave similar results for the order of 10-100 results as was reported by Liu et al. [47], and found it to be congruent with those. We discuss further in chapter 6.

QA-NER	**Prompt**	**Zero-shot**	**1-set**	**2-set**
CONLL	1	**0.10**	0.08	0.08
	2	0.22	0.24	**0.27**
	3	0.09	0.08	0.08
	4	**0.19**	0.10	0.10
	5	**0.34**	0.34	0.34
MIT Restaurant	1	0.07	0.00	0.00
	2	**0.17**	0.00	0.02
	3	**0.10**	0.00	0.08
	4	0.17	0.00	**0.18**
	5	**0.27**	0.03	0.18
MIT Movies	1	0.10	0.08	**0.26**
	2	**0.39**	0.27	0.39
	3	0.21	0.05	**0.35**
	4	0.28	0.08	**0.34**
	5	**0.48**	0.44	0.41

Table 5.2: F1 scores for the QA-NER reimplementation with bert-large-squad2 across the unimodal datasets. Scores are shown for corresponding prompt and few shot examples. In this case the examples are the traditional sets, so N-times each entity type. The best results for each row are bolded. Best for the dataset is underlined.

5.2 Language datasets

The experiment on language datasets, CONLL, MIT Restaurant and MIT Movie, are presented here. For each dataset, we present tables showing the results across all models and prompt variations, according to the number of in-context examples.

5.2.1 CONLL

Results for the CONLL-2003 dataset with the stringent metric are presented in Table 5.3. In the appendix, Table D9 shows the relative difference between the multimodal models and their underlying LLMs.

5.2.2 MiT Restaurant

Results for the MiT Restaurant dataset with the stringent metric are presented in Table 5.4. In the appendix, Table D10 shows the relative difference between the multimodal models and their underlying LLM's

5.2.3 MiT Movie

Results for the MiT Movie dataset with the stringent metric are presented in Table 5.5. In the appendix, Table D11 shows the relative difference between the multimodal models and their underlying LLM's

5.3 Multimodal datasets

As in the previous section, the results for each dataset (Twitter2015 and Twitter2017) are presented for all models and prompt variations. In addition, the results with and without using images from the dataset are presented for the comparative difference in results. We do this by providing the results as the relative difference with and without images, the absolute results with images can be found in appendix section D.

5.3.1 Twitter2015

Results for the Twitter2015 dataset with the stringent metric are presented in Table 5.6. Further the relative difference with regards to using the image from

Model	Prompt	Zero-shot	1-shot	2-shot	3-shot	4-shot/ 1-set	5-shot	6-shot	7-shot	8-shot/ 2-sets
FROMAGe	1	0.00	0.19	0.20	0.20	0.22	0.20	0.19	0.20	0.18
	2	0.01	0.17	0.19	0.20	0.22	0.19	0.19	0.20	0.18
	3	0.00	0.16	0.18	0.19	0.21	0.19	0.20	0.20	0.19
	4	0.01	0.20	0.20	0.21	0.24	0.22	0.20	0.21	0.20
	5	0.02	0.20	0.20	0.21	0.23	0.21	0.20	0.22	0.21
BLIP2-opt-2.7b	1	0.03	0.17	0.09	0.13	0.17	0.22	0.22	0.21	0.21
	2	0.01	0.18	0.07	0.10	0.14	0.15	0.19	0.20	0.20
	3	0.01	0.16	0.09	0.12	0.15	0.19	0.20	0.19	0.20
	4	0.01	0.18	0.11	0.14	0.21	0.23	0.23	0.25	0.24
	5	0.02	0.19	0.09	0.13	0.19	0.22	0.23	0.24	0.24
BLIP2-opt-6.7b	1	0.00	0.22	0.23	0.23	0.24	0.22	0.18	0.19	0.16
	2	0.00	0.23	0.24	0.23	0.25	0.23	0.22	0.24	0.20
	3	0.00	0.23	0.24	0.23	0.24	0.22	0.23	0.22	0.19
	4	0.00	0.22	0.26	0.25	0.25	0.24	0.22	0.22	0.19
	5	0.01	0.24	0.26	0.25	0.26	0.24	0.23	0.23	0.19
BLIP2-flan-t5-xl	1	0.50	0.51	0.52	0.52	0.52	0.51	0.51	0.52	0.50
	2	0.41	0.50	0.52	0.53	0.54	0.53	0.53	0.53	0.51
	3	0.44	0.51	0.53	0.54	0.55	0.53	0.53	0.53	0.50
	4	0.53	0.52	0.54	0.54	0.55	0.53	0.52	0.53	0.51
	5	0.35	0.45	0.49	0.50	0.51	0.49	0.50	0.50	0.49
BLIP2-flan-t5-xxl	1	0.41	0.38	0.41	0.42	0.42	0.41	0.43	0.44	0.43
	2	0.51	**0.55**	0.55	**0.56**	0.55	0.53	0.55	0.54	0.52
	3	0.49	0.51	0.52	0.52	0.52	0.50	0.51	0.51	0.50
	4	0.47	0.50	0.50	0.51	0.50	0.50	0.50	0.50	0.50
	5	0.47	0.49	0.51	0.52	0.52	0.51	0.52	0.52	0.51
mPLUG-Owl	1	0.29	0.25	0.28	0.28	0.28	0.27	0.28	0.27	0.28
	2	0.27	0.27	0.30	0.29	0.29	0.29	0.29	0.29	0.30
	3	0.22	0.24	0.28	0.26	0.27	0.27	0.26	0.27	0.29
	4	0.27	0.27	0.29	0.31	0.29	0.29	0.31	0.30	0.32
	5	0.29	0.27	0.30	0.30	0.30	0.30	0.30	0.32	0.32
OpenFlamingo	1	0.11	0.19	0.18	0.17	0.21	0.19	0.20	0.22	0.22
	2	0.06	0.08	0.12	0.10	0.13	0.11	0.13	0.14	0.15
	3	0.07	0.07	0.08	0.08	0.10	0.10	0.10	0.12	0.12
	4	0.11	0.17	0.19	0.17	0.21	0.20	0.20	0.23	0.23
	5	0.07	0.14	0.17	0.14	0.18	0.18	0.18	0.21	0.21
FlanT5-XL	1	0.53	0.50	0.52	0.51	0.53	0.51	0.51	0.51	0.50
	2	0.48	0.49	0.51	0.52	0.54	0.53	0.53	0.53	0.51
	3	0.53	0.50	0.52	0.53	0.54	0.53	0.52	0.53	0.50
	4	**0.55**	0.51	0.52	0.54	0.54	0.53	0.53	0.52	0.51
	5	0.47	0.44	0.50	0.50	0.51	0.49	0.51	0.50	0.49
FlanT5-XXL	1	0.41	0.37	0.41	0.41	0.42	0.41	0.42	0.44	0.42
	2	0.52	0.54	0.54	0.55	0.54	0.52	0.54	0.54	0.51
	3	0.47	0.50	0.50	0.50	0.50	0.49	0.50	0.50	0.49
	4	0.45	0.49	0.49	0.50	0.49	0.49	0.49	0.50	0.49
	5	0.46	0.47	0.50	0.51	0.51	0.50	0.50	0.51	0.50
OPT-2.7B	1	0.03	0.18	0.20	0.19	0.20	0.22	0.20	0.22	0.22
	2	0.04	0.17	0.20	0.20	0.21	0.21	0.20	0.22	0.22
	3	0.06	0.15	0.18	0.17	0.20	0.21	0.20	0.21	0.20
	4	0.05	0.16	0.21	0.20	0.23	0.23	0.22	0.24	0.23
	5	0.03	0.16	0.21	0.19	0.22	0.22	0.22	0.23	0.23
OPT-6.7B	1	0.00	0.14	0.20	0.18	0.21	0.20	0.20	0.21	0.20
	2	0.00	0.14	0.20	0.17	0.21	0.19	0.21	0.21	0.20
	3	0.00	0.10	0.17	0.16	0.20	0.19	0.20	0.20	0.19
	4	0.00	0.16	0.20	0.18	0.22	0.22	0.22	0.23	0.23
	5	0.00	0.18	0.22	0.19	0.22	0.22	0.22	0.24	0.23
LLAMA-7B	1	0.25	0.24	0.27	0.26	0.26	0.27	0.28	0.29	0.29
	2	0.16	0.25	0.27	0.27	0.27	0.27	0.27	0.29	0.29
	3	0.16	0.22	0.25	0.24	0.25	0.26	0.26	0.28	0.27
	4	0.26	0.27	0.30	0.30	0.30	0.30	0.30	0.31	0.31
	5	0.25	0.27	0.29	0.30	0.30	0.30	0.31	0.31	0.32

Table 5.3: Mean Micro-F1 scores for the CONLL dataset using the stringent evaluation metric. Score averages are for models with their corresponding prompts and amount of random N-singular few-shot examples. Selected high scores are bolded, best score for each column is underlined.

Model	Prompt	Zero-shot	1-shot	2-shot	3-shot	4-shot	5-shot	6-shot	7-shot	8-shot/ 1-set
FROMAGe	1	0.01	0.07	0.07	0.09	0.08	0.08	0.08	0.08	0.09
	2	0.02	0.06	0.07	0.08	0.09	0.08	0.07	0.07	0.08
	3	0.00	0.05	0.07	0.08	0.09	0.09	0.07	0.07	0.08
	4	0.03	0.06	0.06	0.08	0.09	0.08	0.07	0.07	0.09
	5	0.03	0.06	0.05	0.07	0.09	0.08	0.08	0.07	0.09
BLIP2-opt-2.7b	1	0.01	0.09	0.05	0.05	0.06	0.09	0.09	0.09	0.09
	2	0.02	0.09	0.04	0.05	0.03	0.07	0.07	0.07	0.08
	3	0.00	0.08	0.03	0.05	0.04	0.05	0.05	0.06	0.07
	4	0.00	0.08	0.03	0.04	0.05	0.08	0.07	0.07	0.08
	5	0.02	0.09	0.03	0.05	0.06	0.07	0.08	0.09	0.10
BLIP2-opt-6.7b	1	0.00	0.09	0.09	0.11	0.11	0.11	0.11	0.10	0.10
	2	0.01	0.09	0.10	0.10	0.12	0.11	0.11	0.10	0.10
	3	0.00	0.09	0.10	0.10	0.11	0.11	0.10	0.09	0.10
	4	0.00	0.10	0.10	0.11	0.13	0.12	0.10	0.10	0.11
	5	0.00	0.10	0.11	0.12	0.13	0.12	0.12	0.11	0.11
BLIP2-flan-t5-xl	1	0.24	0.30	0.30	0.31	0.30	0.31	0.30	0.31	0.30
	2	0.19	0.35	0.33	0.35	0.33	0.34	0.35	0.34	0.33
	3	0.22	0.30	0.32	0.32	0.32	0.32	0.33	0.32	0.32
	4	0.24	0.29	0.29	0.30	0.30	0.30	0.32	0.32	0.32
	5	0.26	0.34	0.34	0.35	0.34	0.35	0.35	0.35	0.35
BLIP2-flan-t5-xxl	1	0.30	0.31	0.33	0.32	0.32	0.33	0.32	0.33	0.33
	2	0.37	**0.44**	0.41	0.42	0.42	0.42	0.41	0.41	0.42
	3	0.36	0.40	0.39	0.41	0.41	0.40	0.40	0.40	0.40
	4	**0.39**	0.41	0.40	0.41	0.42	0.42	0.42	0.42	0.41
	5	0.37	0.41	0.39	0.41	0.41	0.41	0.42	0.41	0.42
mPLUG-Owl	1	0.17	0.13	0.13	0.14	0.14	0.15	0.14	0.14	0.13
	2	0.14	0.13	0.17	0.18	0.18	0.17	0.17	0.16	0.16
	3	0.15	0.12	0.15	0.15	0.15	0.15	0.15	0.15	0.13
	4	0.16	0.13	0.16	0.16	0.16	0.16	0.15	0.14	0.14
	5	0.21	0.20	0.20	0.21	0.21	0.22	0.20	0.19	0.19
OpenFlamingo	1	0.13	0.12	0.13	0.15	0.15	0.15	0.14	0.14	0.14
	2	0.11	0.10	0.10	0.13	0.13	0.14	0.12	0.13	0.13
	3	0.10	0.09	0.11	0.13	0.14	0.14	0.13	0.12	0.13
	4	0.10	0.11	0.12	0.14	0.15	0.15	0.12	0.12	0.13
	5	0.11	0.14	0.15	0.18	0.18	0.17	0.15	0.15	0.15
FlanT5-XL	1	0.24	0.30	0.29	0.31	0.30	0.32	0.30	0.30	0.30
	2	0.23	0.36	0.35	0.36	0.35	0.36	0.36	0.35	0.35
	3	0.25	0.31	0.33	0.35	0.33	0.34	0.34	0.33	0.33
	4	0.25	0.32	0.32	0.32	0.31	0.31	0.34	0.33	0.32
	5	0.28	0.34	0.36	0.36	0.36	0.36	0.36	0.35	0.35
FlanT5-XXL	1	0.30	0.31	0.32	0.32	0.32	0.32	0.32	0.33	0.33
	2	0.37	**0.45**	0.42	0.42	0.43	0.42	0.43	0.42	0.43
	3	0.36	0.40	0.39	0.40	0.41	0.40	0.41	0.41	0.40
	4	**0.38**	0.42	0.42	0.42	0.43	0.42	0.42	0.42	0.42
	5	0.36	**0.44**	0.42	0.42	0.42	0.42	0.43	0.42	0.43
OPT-2.7B	1	0.01	0.06	0.10	0.10	0.10	0.10	0.09	0.09	0.10
	2	0.02	0.07	0.09	0.11	0.11	0.11	0.10	0.09	0.09
	3	0.01	0.05	0.09	0.10	0.10	0.10	0.09	0.09	0.09
	4	0.01	0.06	0.08	0.10	0.10	0.10	0.09	0.08	0.09
	5	0.01	0.06	0.09	0.11	0.11	0.12	0.10	0.10	0.10
OPT-6.7B	1	0.00	0.07	0.09	0.09	0.11	0.11	0.10	0.09	0.11
	2	0.01	0.06	0.09	0.10	0.11	0.10	0.10	0.08	0.09
	3	0.00	0.05	0.09	0.08	0.10	0.10	0.10	0.08	0.10
	4	0.00	0.07	0.09	0.10	0.11	0.12	0.12	0.11	0.11
	5	0.00	0.07	0.09	0.10	0.12	0.12	0.13	0.11	0.12
LLAMA-7B	1	0.11	0.10	0.11	0.12	0.12	0.12	0.11	0.12	0.12
	2	0.12	0.12	0.13	0.14	0.15	0.15	0.14	0.14	0.14
	3	0.13	0.11	0.11	0.13	0.14	0.14	0.13	0.13	0.14
	4	0.09	0.11	0.12	0.14	0.14	0.14	0.13	0.13	0.13
	5	0.12	0.16	0.16	0.17	0.17	0.17	0.16	0.16	0.16

Table 5.4: Mean Micro-F1 scores for the MIT Restaurant dataset using the stringent evaluation metric. Score averages are for models with their corresponding prompts and amount of random N-singular few-shot examples. Selected high scores are bolded, best score for each column is underlined.

Model	Prompt	Zero-shot	1-shot	2-shot	3-shot	4-shot	5-shot	6-shot	7-shot	8-shot
FROMAGe	1	0.03	0.04	0.07	0.09	0.09	0.09	0.09	0.09	0.09
	2	0.06	0.06	0.09	0.11	0.13	0.11	0.12	0.12	0.11
	3	0.04	0.05	0.08	0.10	0.10	0.09	0.10	0.11	0.10
	4	0.07	0.06	0.10	0.13	0.13	0.12	0.13	0.12	0.10
	5	0.25	0.06	0.11	0.14	0.15	0.14	0.16	0.16	0.14
BLIP2-opt-2.7b	1	0.00	0.01	0.00	0.00	0.01	0.03	0.02	0.05	0.03
	2	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.01	0.02
	3	0.00	0.01	0.00	0.01	0.01	0.01	0.02	0.02	0.01
	4	0.00	0.02	0.00	0.00	0.01	0.01	0.02	0.03	0.04
	5	0.01	0.03	0.01	0.00	0.01	0.01	0.02	0.05	0.06
BLIP2-opt-6.7b	1	0.01	0.05	0.07	0.10	0.11	0.11	0.10	0.09	0.05
	2	0.03	0.08	0.09	0.15	0.12	0.14	0.12	0.11	0.07
	3	0.01	0.06	0.08	0.11	0.10	0.12	0.10	0.08	0.05
	4	0.06	0.05	0.08	0.12	0.13	0.14	0.13	0.12	0.06
	5	0.13	0.05	0.12	0.18	0.18	0.19	0.18	0.18	0.13
BLIP2-flan-t5-xl	1	0.53	0.51	0.52	0.50	0.50	0.49	0.47	0.47	0.47
	2	**0.55**	0.55	0.57	0.57	0.57	0.57	0.57	0.58	**0.59**
	3	**0.56**	0.57	0.58	**0.59**	0.58	0.58	0.58	0.57	**0.59**
	4	**0.57**	0.56	0.57	0.57	0.57	0.58	0.58	0.57	0.58
	5	0.53	0.55	0.55	0.57	0.58	0.58	0.58	**0.59**	**0.59**
BLIP2-flan-t5-xxl	1	0.39	0.38	0.42	0.41	0.41	0.42	0.42	0.42	0.42
	2	0.51	0.52	0.52	0.53	0.53	0.53	0.52	0.52	0.52
	3	0.44	0.47	0.50	0.50	0.50	0.50	0.49	0.49	0.48
	4	0.47	0.50	0.52	0.53	0.52	0.52	0.52	0.51	0.51
	5	0.49	0.52	0.52	0.54	0.53	0.54	0.53	0.53	0.52
mPLUG-Owl	1	0.27	0.07	0.16	0.18	0.15	0.15	0.17	0.18	0.16
	2	0.39	0.06	0.21	0.22	0.21	0.23	0.22	0.20	0.20
	3	0.37	0.07	0.16	0.19	0.18	0.20	0.19	0.18	0.18
	4	0.39	0.08	0.20	0.22	0.20	0.21	0.20	0.20	0.18
	5	0.45	0.12	0.27	0.29	0.27	0.29	0.27	0.28	0.26
OpenFlamingo	1	0.04	0.04	0.06	0.11	0.11	0.08	0.11	0.13	0.10
	2	0.16	0.04	0.05	0.07	0.04	0.03	0.04	0.05	0.04
	3	0.12	0.02	0.05	0.07	0.06	0.03	0.06	0.06	0.06
	4	0.12	0.04	0.07	0.08	0.06	0.03	0.06	0.08	0.07
	5	0.17	0.05	0.08	0.10	0.07	0.05	0.07	0.09	0.08
FlanT5-XL	1	0.51	0.49	0.50	0.49	0.49	0.47	0.46	0.46	0.47
	2	0.56	0.54	0.55	0.55	0.55	0.55	0.56	0.56	0.57
	3	0.57	0.55	0.56	0.56	0.56	0.57	0.56	0.56	0.57
	4	**0.59**	0.56	0.57	**0.58**	0.57	0.58	0.58	0.58	0.58
	5	0.56	0.53	0.55	0.57	0.58	0.58	0.58	0.58	0.58
FlanT5-XXL	1	0.39	0.37	0.42	0.42	0.42	0.43	0.43	0.43	0.42
	2	0.51	0.51	0.53	0.53	0.54	0.53	0.52	0.52	0.52
	3	0.48	0.48	0.51	0.51	0.51	0.50	0.51	0.50	0.49
	4	0.50	0.51	0.53	0.53	0.53	0.53	0.53	0.52	0.51
	5	0.52	0.51	0.53	0.54	0.54	0.54	0.54	0.53	0.52
OPT-2.7B	1	0.01	0.05	0.05	0.07	0.06	0.06	0.07	0.07	0.07
	2	0.07	0.05	0.07	0.10	0.09	0.08	0.08	0.10	0.09
	3	0.01	0.04	0.06	0.08	0.07	0.07	0.07	0.09	0.08
	4	0.05	0.04	0.07	0.09	0.08	0.07	0.08	0.08	0.08
	5	0.07	0.05	0.09	0.12	0.11	0.10	0.11	0.12	0.10
OPT-6.7B	1	0.01	0.05	0.06	0.08	0.07	0.08	0.08	0.10	0.08
	2	0.02	0.07	0.10	0.12	0.12	0.13	0.12	0.12	0.10
	3	0.02	0.05	0.08	0.10	0.10	0.11	0.10	0.10	0.09
	4	0.01	0.05	0.07	0.09	0.09	0.10	0.10	0.11	0.09
	5	0.01	0.07	0.09	0.11	0.12	0.13	0.12	0.14	0.11
LLAMA-7B	1	0.13	0.04	0.08	0.12	0.13	0.12	0.14	0.15	0.14
	2	0.22	0.05	0.14	0.16	0.14	0.16	0.15	0.14	0.14
	3	0.18	0.05	0.10	0.15	0.14	0.16	0.15	0.15	0.14
	4	0.26	0.05	0.13	0.15	0.14	0.15	0.15	0.16	0.15
	5	0.30	0.07	0.19	0.20	0.19	0.21	0.19	0.20	0.20

Table 5.5: Mean Micro-F1 scores for the MIT Movie dataset using the stringent evaluation metric. Score averages are for models with their corresponding prompts and amount of random N-singular few-shot examples. Selected high scores are bolded, best score for each column is underlined.

the dataset versus an empty (black) image is presented in Table 5.7. Results when using the images can be found in appendix Table D5

5.3.2 Twitter2017

Results for the Twitter2015 dataset with the stringent metric and not using images are presented in Table 5.8. Further, the relative difference with regards to using the image from the dataset versus an empty (black) image is presented in Table 5.9. Results when using the images can be found in appendix Table D6

5.4 Entity level sampled results and ablations

In this section, we present a detailed sample of the results, where we show the entity level accuracy and ablations. The results are too numerous to generate figures for, so we focus on some of the most performant models. Further sampled results can be found in appendix section D.

5.4.1 Entity level accuracy for the Twitter datasets

5.4.2 Ablations Twitter datasets

Since the Twitter datasets have the same entity types and the same as CONLL, and images we use them for ablations to see both a sample for the effect of instructions without and with visuals. The entity enumeration is ”The entities are persons, locations, organizations, and miscellaneous (other)”.

Instruction ablations			**Twitter2015**		**Twitter2017**	
Intention	**Entity enumeration**	**Image specifier**	**With image**	**Without image**	**With image**	**Without image**
			0.37	0.43	0.34	0.36
		X	0.48	0.45	0.39	0.36
	X		0.44	0.48	0.42	0.44
X			0.51	0.53	0.40	0.42
	X	X	0.50	0.49	0.44	0.44
X	X		0.50	0.53	0.43	0.44
X		X	0.54	0.54	0.43	0.42
X	X	X	0.54	0.54	0.46	0.44

Table 5.10: Ablation over the different instructions for Twitter2015 and Twitter2017. X indicates the inclusion of the relevant instruction. The configuration is Blip2-Flan-T5-XL, with zero-shot and prompt no. 2.

Twitter2015	**Without image**									
Model	**Prompt**	**Zero-shot**	**1-shot**	**2-shot**	**3-shot**	**4-shot/ 1-set**	**5-shot**	**6-shot**	**7-shot**	**8-shot/ 2-sets**
FROMAGe	1	0.01	0.13	0.16	0.17	0.14	0.15	0.16	0.17	**0.18**
	2	0.00	0.10	0.13	0.15	0.15	0.14	**0.16**	0.14	0.16
	3	0.00	0.10	0.13	**0.15**	0.15	0.14	0.14	0.13	0.14
	4	0.01	0.16	0.16	0.18	0.17	0.17	0.19	0.18	**0.20**
	5	0.02	0.14	0.16	**0.20**	0.18	0.18	0.20	0.19	0.20
BLIP2-opt-2.7b	1	0.01	0.08	0.11	0.12	0.12	0.13	**0.15**	0.13	0.13
	2	0.01	0.07	0.09	0.10	0.13	0.14	0.16	0.16	**0.18**
	3	0.00	0.07	0.11	0.12	0.13	0.12	0.13	0.15	**0.16**
	4	0.01	0.09	0.13	0.12	0.13	0.13	0.15	0.16	**0.17**
	5	0.02	0.10	0.12	0.12	0.12	0.12	0.14	**0.15**	0.15
BLIP2-opt-6.7b	1	0.00	0.11	0.12	0.17	0.17	0.17	0.23	**0.26**	0.26
	2	0.00	0.13	0.11	0.19	0.22	0.20	0.21	**0.25**	0.22
	3	0.00	0.14	0.11	0.20	0.20	0.20	0.21	**0.23**	0.23
	4	0.00	0.15	0.15	0.19	0.18	0.17	0.20	**0.22**	0.22
	5	0.00	0.16	0.14	0.20	0.19	0.18	0.22	**0.24**	0.24
BLIP2-flan-t5-xl	1	**0.60**	0.60	0.57	0.55	0.56	0.56	0.57	0.54	0.55
	2	0.54	**0.59**	0.59	0.57	0.57	0.57	0.57	0.53	0.53
	3	0.60	**0.61**	0.59	0.57	0.57	0.57	0.57	0.54	0.55
	4	**0.63**	0.60	0.60	0.56	0.57	0.57	0.57	0.54	0.54
	5	0.52	**0.59**	0.59	0.57	0.57	0.57	0.56	0.53	0.54
BLIP2-flan-t5-xxl	1	0.40	0.38	0.39	0.35	0.39	0.41	**0.43**	0.40	0.42
	2	0.59	**0.60**	0.57	0.52	0.53	0.52	0.53	0.50	0.50
	3	**0.53**	0.51	0.50	0.49	0.50	0.50	0.50	0.50	0.50
	4	0.50	0.49	**0.50**	0.49	0.50	0.50	0.50	0.50	0.50
	5	**0.54**	0.52	0.51	0.49	0.51	0.51	0.51	0.50	0.50
mPLUG-Owl	1	0.30	0.22	0.27	0.30	**0.32**	0.32	0.31	0.31	0.30
	2	0.31	0.32	0.32	0.32	0.33	**0.34**	0.33	0.32	0.31
	3	0.27	0.30	0.29	0.30	0.30	**0.32**	0.31	0.29	0.30
	4	0.34	0.29	0.31	0.33	0.33	0.34	**0.35**	0.33	0.34
	5	0.34	**0.36**	0.34	0.34	0.36	0.36	0.36	0.34	0.34
OpenFlamingo	1	0.11	0.21	0.27	0.32	0.31	**0.33**	0.31	0.28	0.28
	2	0.11	0.20	0.24	0.25	**0.28**	0.27	0.25	0.19	0.19
	3	0.09	0.20	0.23	0.25	**0.27**	0.27	0.23	0.19	0.20
	4	0.12	0.23	0.29	0.31	0.32	**0.33**	0.32	0.29	0.27
	5	0.15	0.27	0.32	0.34	0.33	**0.36**	0.34	0.32	0.30

Table 5.6: Mean Micro-F1 scores for the Twitter2015 dataset using the stringent evaluation metric when using empty (black) images. Score averages are for models with their corresponding prompts and amount of random N-singular few-shot examples. The highest scores for each row are bolded with a preference for smaller amounts of training data, the best score for each column is underlined.

Twitter2015	**Difference in score between with image and without image**									
Model	**Prompt**	**Zero-shot**	**1-shot**	**2-shot**	**3-shot**	**4-shot/ 1-set**	**5-shot**	**6-shot**	**7-shot**	**8-shot/ 2-sets**
FROMAGe	1	-0.01	-0.04	-0.04	-0.03	-0.01	-0.02	-0.03	-0.02	-0.01
	2	0.00	-0.01	0.01	-0.01	0.00	0.02	-0.01	0.01	0.01
	3	0.01	-0.00	-0.00	-0.00	-0.01	0.02	0.01	0.01	0.02
	4	-0.00	-0.03	-0.02	-0.01	-0.02	-0.00	-0.01	0.00	-0.01
	5	0.01	-0.02	-0.00	-0.03	-0.02	-0.02	-0.02	-0.01	-0.00
BLIP2-opt-2.7b	1	0.01	-0.01	-0.01	-0.01	0.00	0.01	0.01	0.02	0.02
	2	0.01	-0.01	-0.01	0.01	-0.01	0.01	0.01	0.01	-0.00
	3	0.01	-0.01	-0.01	-0.00	0.00	0.02	0.03	0.02	0.01
	4	0.02	-0.01	-0.01	0.01	-0.01	0.02	0.01	0.02	0.01
	5	0.01	-0.02	-0.01	0.01	0.00	0.03	0.03	0.02	0.02
BLIP2-opt-6.7b	1	0.01	-0.02	0.03	0.00	0.03	0.02	-0.01	-0.05	-0.06
	2	0.00	-0.02	0.01	-0.02	-0.02	-0.02	-0.01	-0.07	-0.05
	3	0.00	-0.03	0.02	-0.03	-0.02	-0.02	-0.03	-0.07	-0.06
	4	0.01	-0.01	0.03	-0.00	0.02	0.02	0.01	-0.03	-0.03
	5	0.03	-0.03	0.03	0.01	0.01	0.02	-0.01	-0.03	-0.04
BLIP2-flan-t5-xl	1	0.00	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02
	2	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.01
	3	0.01	-0.01	-0.00	-0.01	-0.01	-0.00	0.00	-0.01	-0.01
	4	0.01	-0.01	-0.01	-0.01	-0.01	0.01	-0.00	-0.00	0.00
	5	-0.02	-0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02
BLIP2-flan-t5-xxl	1	-0.00	0.01	-0.01	0.01	-0.02	-0.02	-0.02	-0.00	-0.02
	2	-0.01	0.00	-0.01	-0.01	-0.01	-0.00	-0.01	-0.01	-0.00
	3	-0.01	-0.00	-0.00	-0.00	-0.01	-0.00	-0.01	0.00	0.00
	4	0.00	-0.01	-0.01	-0.00	-0.00	-0.01	-0.01	-0.00	-0.01
	5	-0.03	-0.00	-0.01	-0.00	-0.01	-0.01	-0.01	-0.00	0.00
mPLUG-Owl	1	0.00	0.00	0.01	0.02	0.00	0.02	0.01	0.01	0.01
	2	-0.02	0.01	-0.01	0.00	-0.00	-0.00	0.01	0.01	-0.00
	3	-0.01	0.01	-0.00	0.01	0.01	-0.01	0.00	0.00	-0.01
	4	0.02	0.00	0.01	0.01	0.00	0.02	0.01	0.01	0.01
	5	0.03	-0.01	0.01	0.01	0.01	0.02	0.01	0.01	-0.00
OpenFlamingo	1	0.02	-0.01	0.01	-0.00	-0.00	0.00	0.02	0.02	0.01
	2	0.04	0.02	0.00	0.01	-0.01	-0.00	-0.02	0.01	-0.00
	3	0.05	0.01	0.02	0.00	-0.01	0.00	0.01	0.00	-0.02
	4	0.00	0.02	0.03	0.01	0.01	0.00	0.01	-0.01	-0.00
	5	-0.04	-0.00	0.01	-0.00	0.00	-0.01	0.01	0.00	0.01

Table 5.7: Difference in Micro-F1 scores for the Twitter2015 dataset using the stringent evaluation metric when using the dataset's images and subtracting the scores for when using an empty (black) image. Scores averages are for models with their corresponding prompts and amount of random N-singular few-shot examples.

Twitter2017	**Without image**									
Model	**Prompt**	**Zero-shot**	**1-shot**	**2-shot**	**3-shot**	**4-shot/ 1-set**	**5-shot**	**6-shot**	**7-shot**	**8-shot/ 2-sets**
FROMAGe	1	0.00	0.19	0.19	0.20	**0.21**	0.16	0.17	0.15	0.14
	2	0.01	0.18	0.15	**0.20**	0.18	0.14	0.14	0.15	0.15
	3	0.00	0.17	0.15	**0.20**	0.18	0.14	0.15	0.15	0.15
	4	0.00	**0.22**	0.18	0.21	0.22	0.15	0.18	0.16	0.16
	5	0.02	**0.22**	0.18	0.21	0.22	0.15	0.17	0.17	0.16
BLIP2-opt-2.7b	1	0.01	0.14	0.01	0.02	**0.18**	0.14	0.14	0.12	0.15
	2	0.02	0.16	0.01	0.02	0.08	0.14	0.15	0.17	**0.18**
	3	0.01	0.15	0.01	0.02	0.08	0.13	0.13	0.15	**0.19**
	4	0.02	0.13	0.04	0.06	0.15	0.18	0.19	**0.20**	0.18
	5	0.03	0.18	0.02	0.06	0.14	**0.19**	0.19	0.19	0.18
BLIP2-opt-6.7b	1	0.00	0.24	0.25	**0.26**	0.26	0.24	0.24	0.24	0.26
	2	0.00	0.26	0.25	0.25	**0.27**	0.25	0.25	0.25	0.26
	3	0.00	0.22	0.23	0.24	**0.26**	0.25	0.24	0.24	0.26
	4	0.00	**0.29**	0.29	0.27	0.29	0.26	0.26	0.25	0.28
	5	0.00	**0.30**	0.30	0.28	0.28	0.27	0.26	0.26	0.28
BLIP2-flan-t5-xl	1	**0.59**	0.57	0.56	0.56	0.55	0.56	0.55	0.55	0.55
	2	0.44	**0.51**	0.51	0.51	0.51	0.51	0.51	0.51	0.51
	3	0.54	**0.55**	0.55	0.54	0.54	0.54	0.54	0.54	0.54
	4	**0.58**	0.57	0.55	0.55	0.55	0.55	0.54	0.55	0.54
	5	0.45	0.51	0.51	0.50	0.51	0.51	**0.52**	0.51	0.51
BLIP2-flan-t5-xxl	1	0.42	0.36	0.46	**0.47**	0.46	0.47	0.44	0.45	0.45
	2	**0.60**	0.56	0.54	0.54	0.53	0.53	0.53	0.54	0.54
	3	**0.54**	0.51	0.53	0.53	0.54	0.54	0.53	0.53	0.53
	4	0.51	0.51	**0.53**	0.53	0.53	0.53	0.53	0.53	0.53
	5	**0.53**	0.49	0.51	0.51	0.52	0.52	0.52	0.52	0.52
mPLUG-Owl	1	0.29	0.30	0.29	**0.31**	0.31	0.26	0.27	0.27	0.28
	2	0.27	0.26	0.23	**0.28**	0.28	0.24	0.25	0.26	0.27
	3	0.22	**0.28**	0.25	0.27	0.27	0.23	0.24	0.24	0.26
	4	0.32	0.32	0.29	0.32	**0.33**	0.27	0.30	0.31	0.31
	5	0.32	0.30	0.27	0.31	**0.34**	0.28	0.31	0.32	0.32
OpenFlamingo	1	0.09	0.29	0.24	0.30	0.31	**0.32**	0.32	0.31	0.32
	2	0.09	0.14	0.10	0.17	0.22	**0.28**	0.27	0.25	0.26
	3	0.08	0.16	0.15	0.22	0.25	**0.28**	0.28	0.26	0.27
	4	0.10	**0.33**	0.27	0.31	0.32	0.32	0.32	0.32	0.33
	5	0.10	0.31	0.20	0.29	0.31	**0.33**	0.33	0.32	0.32

Table 5.8: Mean Micro-F1 scores for the Twitter2017 dataset using the stringent evaluation metric when using empty (black) images. Score averages are for models with their corresponding prompts and amount of random N-singular few-shot examples. The highest scores for each row are bolded with a preference for smaller amounts of training data, the best score for each column is underlined.

Twitter2017	Difference in score between with image and without image									
Model	**Prompt**	**Zero-shot**	**1-shot**	**2-shot**	**3-shot**	**4-shot/ 1-set**	**5-shot**	**6-shot**	**7-shot**	**8-shot/ 2-sets**
FROMAGe	1	0.00	-0.10	-0.04	-0.01	-0.04	0.04	0.03	0.05	0.03
	2	0.00	-0.10	-0.02	-0.01	0.00	0.04	0.05	0.02	0.04
	3	0.01	-0.09	-0.03	-0.02	-0.00	0.06	0.05	0.03	0.04
	4	0.00	-0.09	-0.02	0.00	-0.03	0.06	0.03	0.04	0.04
	5	-0.00	-0.09	-0.03	-0.01	-0.02	0.05	0.05	0.06	0.06
BLIP2-opt-2.7b	1	0.01	-0.03	0.07	0.08	-0.01	0.00	0.01	0.03	-0.02
	2	-0.01	-0.08	0.03	0.03	0.01	0.00	-0.03	-0.04	-0.01
	3	-0.00	-0.08	0.03	0.03	0.01	0.00	-0.00	-0.01	-0.02
	4	0.01	-0.00	0.02	0.04	-0.00	-0.03	-0.05	-0.03	-0.03
	5	0.02	-0.07	0.03	0.02	0.02	-0.06	-0.04	-0.01	-0.03
BLIP2-opt-6.7b	1	0.00	-0.09	-0.05	-0.04	-0.04	-0.05	-0.06	0.01	-0.00
	2	0.00	-0.09	-0.07	-0.03	-0.03	-0.03	-0.05	0.00	0.01
	3	0.00	-0.07	-0.05	-0.02	-0.03	-0.03	-0.04	0.01	0.01
	4	0.02	-0.10	-0.06	-0.02	-0.03	-0.01	-0.04	0.02	0.00
	5	0.02	-0.09	-0.08	-0.04	-0.02	-0.01	-0.03	0.01	0.01
BLIP2-flan-t5-xl	1	-0.03	-0.02	-0.01	-0.00	0.00	0.00	0.00	0.00	0.01
	2	0.02	-0.01	-0.00	0.00	-0.00	0.00	0.00	0.00	0.00
	3	0.01	-0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.00
	4	-0.02	0.00	0.01	0.00	-0.00	0.00	0.01	0.00	0.01
	5	-0.04	0.01	0.00	0.01	-0.00	0.00	-0.00	0.00	0.01
BLIP2-flan-t5-xxl	1	-0.01	0.01	-0.02	-0.03	-0.03	-0.04	-0.04	-0.05	-0.04
	2	-0.03	-0.02	-0.01	-0.01	-0.01	-0.01	-0.01	-0.02	-0.02
	3	-0.02	-0.01	-0.01	-0.01	-0.02	-0.02	-0.02	-0.01	-0.02
	4	-0.03	-0.02	-0.02	-0.02	-0.01	-0.01	-0.02	-0.01	-0.01
	5	-0.03	-0.01	-0.01	-0.01	-0.02	-0.02	-0.02	-0.02	-0.02
mPLUG-Owl	1	0.01	-0.00	0.01	0.00	0.01	0.01	0.01	0.02	0.01
	2	0.00	0.01	0.00	-0.00	-0.01	-0.00	0.01	0.01	-0.00
	3	0.02	-0.01	-0.01	-0.01	0.00	-0.00	0.00	0.00	0.00
	4	0.02	0.00	0.00	0.00	-0.00	0.01	0.01	0.02	0.02
	5	0.01	0.00	0.02	0.01	0.00	0.01	0.01	0.02	0.02
OpenFlamingo	1	0.04	0.02	0.02	0.01	0.00	0.01	-0.00	0.00	-0.01
	2	0.04	0.04	0.04	0.02	0.02	-0.00	0.01	-0.00	-0.01
	3	0.04	0.05	0.01	0.00	0.02	0.00	-0.00	-0.01	0.00
	4	0.03	-0.00	0.01	0.00	0.01	0.01	0.00	-0.00	-0.00
	5	0.01	0.02	0.03	0.01	0.00	0.00	-0.00	-0.01	-0.01

Table 5.9: Difference in Micro-F1 scores for the Twitter2017 dataset using the stringent evaluation metric when using the dataset's images and subtracting the scores for when using an empty (black) image. Scores averages are for models with their corresponding prompts and amount of random N-singular few-shot examples.

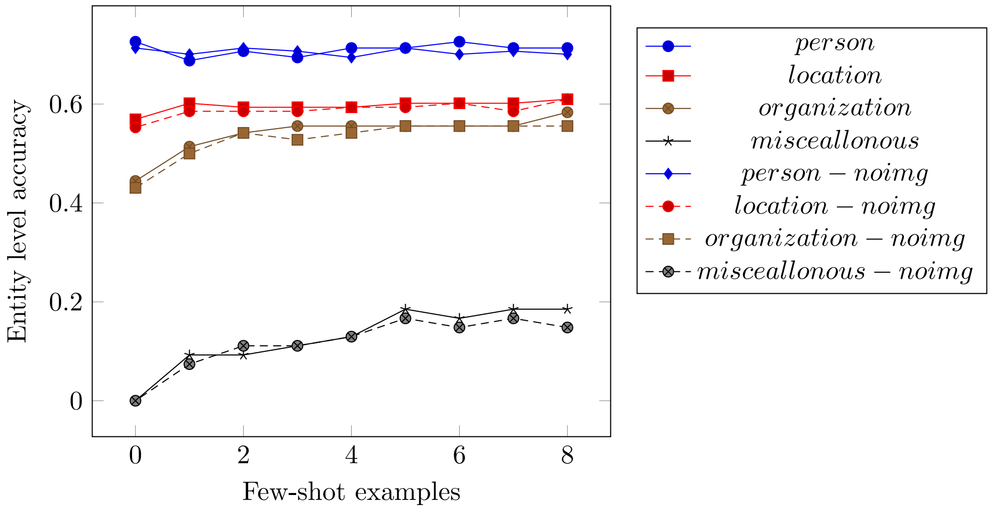

Figure 5.1: Accuracy per entity for Twitter2015 with BLIP2-Flan-T5-XL using prompt no. 2 with and without images. With images is the original in the legend, while without is the specified by ”noimg”. We observe the relative changes in entity accuracy. This also generalizes across most of the results using Flan-T5 models.

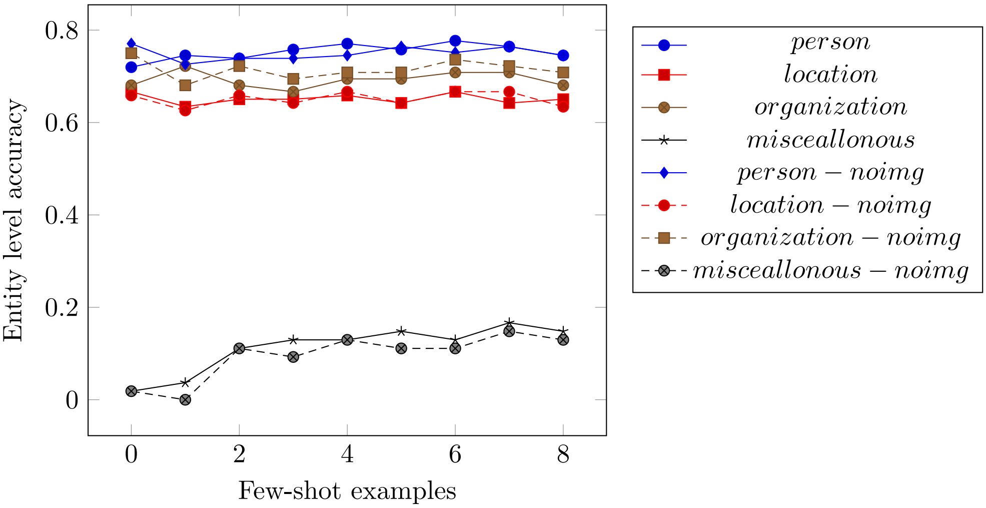

Figure 5.2: Accuracy per entity for Twitter2015 with BLIP2-Flan-T5-XXL using prompt no. 2 with and without images. With images is the original in the legend, while without is the specified by ”noimg”. We observe the relative changes in entity accuracy. This also generalizes across most of the results using Flan-T5 models.

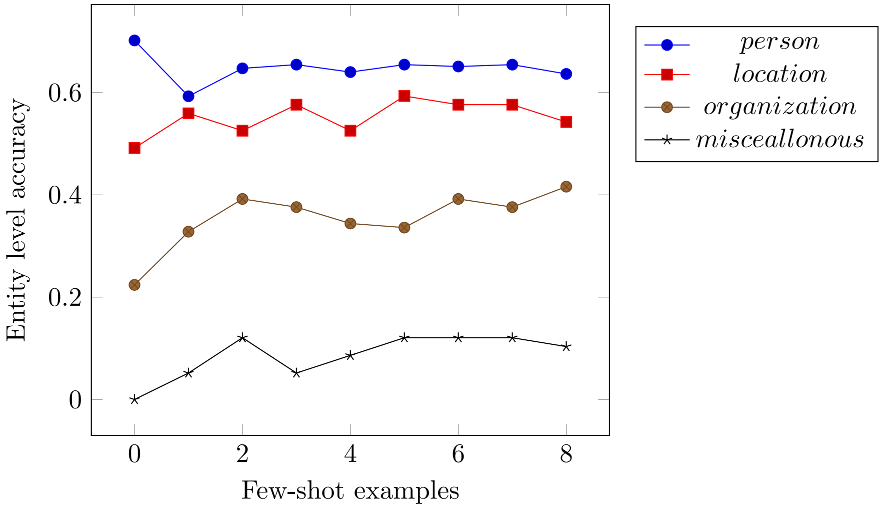

Figure 5.3: Accuracy per entity for Twitter2017 with Flan-T5-XL using prompt no. 3 without images. With images is the original in the legend, while without is the specified by ”noimg”. We observe the relative changes in entity accuracy.

Chapter 6

Discussion

With the presented results in chapter 5, we interpret them with our research questions in mind and relative to performance in other research. We include surveyed results from other research that use techniques that prohibit re-implementation with either our methods or our very low-resource training data context, but from which we can reflect on the comparative performance.

The chapter is sectioned by first discussing the validity of the results, then the results using only language over all the datasets since the approach is identical to multimodal without images. This is the discussion of prompting generative models with only language. We then discuss the results obtained when integrating images and how they compare to only using language. Finally, we discuss the results reflecting on the underlying data and the evaluation metrics.

6.1 Validity

The large amount of data points (F1 scores) generated by the experiment showcases the impact of slight alterations of factors on the results. We see that for the baseline, only slight fine-tuning can cause a sudden drop, as mentioned by Mosbach et al. [58], fine tuning can be quite unstable process. Still, our initial zero-shot results are in line with the reported results by Liu et al. [47].

What becomes more pronounced is the wildly varying results from changing small parts of the prompts for the generative models. The reliance on off-the-shelf models that may have instabilities themselves or covert overriding parameters. in addition to the ability to rely on the multimodal models that are fed empty images versus the LLM, they are based on. We see clear discrepancies between the LLM and the multimodal model, and it makes the prospect of attributing any relative gain to using the actual image from the Twitter datasets somewhat harder. Still, the relative gain should, as far as we know only be impacted by the inclusion of the image versus an empty image, and all models do run stable and deterministically at zero-shot experiments, producing the same outputs.

Additionally, the effect of which training data the pre-trained models have used is not elaborated. And this is due to the fact that it is practically very difficult to elaborate through the large amounts of different datasets for so many pre-trained models and confidently say that there is no relevant NER material in the pre-training data. The mitigating factor for this is that the NER task has not really been rephrased in the way we have and made widespread in natural language form (any more than normal conversation allows for). So that while there probably is useful similar data, the models have been trained on, it's not specifically like what our method takes in and produces.

6.2 Unimodal language results

The results for unimodal language are the most important analysis of how well the generative models can solve named entity recognition as a traditional task. Firstly we discuss the overall performance concerning training data versus other surveyed results. Then we concern ourselves with the varying scores in our results.

6.2.1 Comparative analysis

We include survey results from low-resource or fully fine-tuned research on our language-only datasets to discuss our results compared to other research. Surveyed results for CONLL are presented in Table 6.1. The results use the stringent metric and are not from the same low-resource environment as ours, as the methods use domain transfer from other NER datasets.

As mentioned earlier, fine-tuning or transfer learning from another similar dataset can lead to very good performance, and fully fine-tuned models can reach F1 scores of 90+% on CONLL-2003 [85, 49]. We see that our methods, which have no domain transfer, can achieve results that hover around the 1-shot/set domain transfer results for CONLL-2003 as shown in Table 6.1. Additionally, we mention that COPNER got an F1 score of 46.26 with domain transfer and zero-shot on

Method	1 shot/set	5 shot/sets
CONTaiNER [19]	57.8±5.5	72.8±2.0
ProML [15]	69.16±4.47	79.16±4.49
StructShot [98]	62.4±10.5	74.8±2.4
NNShot [98]	61.2±10.4	74.1±2.3
COPNER [35]	67.0±3.8	74.9±2.9

Table 6.1: Surveyed results for low-resource F1 scores on CONLL-2003, **using domain transfer**. The methods are then trained on another dataset, like OntoNotes v5, before being fine-tuned on CONLL.

Domain transfer / Source	Methods	MIT Restaurant		MIT Movies	
		10-sets	20-sets	10-sets	20-sets
None	Sequence Labeling BERT [17]	21.8	39.4	25.2	42.2
	Template-based BART [17]	46	57.1	37.3	48.5
CONLL-2003	Ziyadi et al. [109]	27.6	29.5	40.1	39.5
	Huang et al. [33]	46.1	48.2	36.4	36.8
	Sequence Labeling BERT [17]	27.2	40.9	28.3	45.2
	Template-based BART [17]	53.1	60.3	42.4	54.2

Table 6.2: Surveyed results for low-resource F1 scores on MIT Restaurant and MIT Movies, using fine-tuning on few samples, or domain transfer on another dataset.

CONLL.

For MIT Restaurant and MIT Movies, we show in Table 6.2 results also using domain transfer, and a couple of results without, but with significantly more training data than we used.

We observe from our best results in Table 5.1 that our methods perform very well, relatively speaking when accounting for the fact that we do not use any explicit domain transfer and that we outperform both the baseline results by a considerable margin and that our methods can produce zero-shot results approaching low-resource domain transfer performance methods for CONLL and MIT Restaurant. And in the case of MIT Movies actually, it actually outperforms the surveyed results with only eight singular training examples, which is less than the number of entity types (less than 1-set).

6.2.2 Performance according to examples

We see that some models, like the Flan-T5 models, perform well on zero-shot but that the F1-metric tapers off with more examples. When looking at the F1-score per entity type, it seems like the additional examples benefit mainly the MISC label and that there is a direct inverse relationship between the MISC label and the overall F1 score (and accuracy). It's also noteworthy that the MISC entity has a lower occurrence in the dataset than the other datasets and that for all our experiments, and most low-resource experiments we have surveyed, the training data few-shot regime is not sampled according to the occurrence, but by including as many of each entity type.

Zero-shot The zero-shot performance of our experiments, where Flan-T5 is used directly or in a multimodal model, is considerably higher than anything we can find in the existing literature. The emphasis of existing literature we find has been on domain transfer from another NER dataset, and not on true low-resource settings.

Many-shots The relative gain in performance for each additional training example varies between the models and across datasets. While the OPT-based models largely start with lower scores and have increasing performance correlated with the number of examples, LLaMA-based models start with moderate zero-shot performance, and the relative gain with examples depends on the data. We see, roughly speaking, that the LLaMA gains more on CONLL and the Twitter datasets, with MIT Restaurant following where the development is not very noticeable, and then MIT Movies where the inclusion of few examples overall decreases performance. This then also matches the number of entities so that the less granular, the more LLaMA was able to leverage the additional singular examples, which could make sense as it then has more relevant examples for each entity type.

Impact of prompts on results

We also observe that the different prompts vary and across both models and data. As a general rule of thumb, prompt no. 2 showed strong performance, being included in several of the best scoring configurations and having few anomalous bad scoring results.

6.2.3 Notes on entities and distribution

We observe some clear, as already noted, variance among the entity distribution. And we see some clear discrepancies in our results concerning predicting what miscellaneous entities are. And while we do think that there are mitigating steps that can be taken, modification of prompting or language used, ”other(s)” could be used instead, for instance. We want to reflect on the entity types set for a dataset.

The task in and of itself is affected by the enumeration of entity types and what entity types are valid. We note that our multimodal dataset has adapted the four entity types from CONLL, which is relatively few entity types, and that the miscellaneous entity class still composes the smallest classified class in the dataset.

It is then interesting that the miscellaneous entity type seems to be the greatest benefiter from adding more examples. And that few-shot training has equally many training examples for each entity type irrespective of the actual distribution.

We also see that there is very little impact of having an enumeration of entities or not. Which for the miscellaneous entity type seems like a confounding fact when the negation of the other entity types defines it.

6.3 Multimodal results integrating visuals

We see a small difference regarding having a black image (no information) vs. an actual image. This difference seems to vary across all factors of the experiment, where we can see that there are almost no rows or columns in either Table 5.7 or Table 5.9 that consistently has only either positive or negative values.

6.3.1 Comparative analysis

We include surveyed results from research on the same multimodal datasets to discuss the results in light of other performing methods.

Surveyed results for Twitter2015 and Twitter2017, where there is a clear differentiation for using the method’s performance with and without images, are presented in Table 6.3. The results use the stringent metric. General results for fully fine-tuned models showing surveyed performance for fully-fine tuned models are shown in Table 6.4.

We see that the relative gain in the surveyed results for using the images has not been very large. Most literature we found has been unable to show more than a 2% higher micro F1 score using visuals. And as such, our results, zero-shot

Method	Twitter2015		Twitter2017	
	Text	Text+Image	Text	Text+Image
ITA-All+CVA [84]	78.25	78.03	89.47	89.75
MRC-MNER [38]	72.74	74.63	85.55	86.85
MNER-QG [37]	72.74	74.94	85.55	87.25

Table 6.3: Surveyed F1 scores for methods showing the results for only text, and for text with the addition of images with the respective datasets.

Modality	Method	Twitter2015	Twitter2017
Text	BERT [37]	71.32	82.95
	BERT-CRF [37]	71.81	83.44
Text + Image	ACoA [86]	70.69	82.15
	ATTR-MMKG-MNER [37]	73.27	
	UMT-BERT-CRF [102]	73.41	85.31
	MAF [94]	73.42	86.25
	RIVA [76]	73.8	
	MRC-MNER [38]	74.63	86.85
	RpBERT [77]	74.8	85.51
	UMGF [103]	74.85	85.51
	MNER-QG [37]	74.94	87.25
	ITA-All+CVA [84]	78.03	89.75
	PromptMNER [86]	78.6	90.27

Table 6.4: Surveyed F1 scores for methods fully-fine-tuned on Twitter2015 and Twitter2017.

results, for example, show the largest relative differences. But this is not necessarily very noteworthy, as the results generally have large volatility regarding which model, data, and amount of training examples are used, even by just examining zero-shot performance across the Twitter datasets, which can be found in Table 5.9 and Table 5.7, we can not find a model which is consistently able to show gains irrespective of which prompt is used. And it's quite visible that smaller alterations in the prompt can change the relative gains displayed wildly. Still, we can display that it's possible to arrange circumstances where experiments will show relative gains larger than the existing literature by relative gain over an empty (black) image. When we compare the results with the underlying LLMs, the relative differences are circumstantial, meaning that they are not consistent and vary in the same fashion.

The lack of low-resource MNER means that it is also hard to evaluate performance relative to other research since it mostly emphasizes tuning with all available training data. Still, our methods, as shown for unimodal datasets, also perform what we would call relatively well compared to the fully tuned models in Table 6.4, since we are able to reach around 60% F1 scores with trivially small amounts of data, compared to the current state-of-the-art results towering around 70+% for Twitter2015 and 80-90% for Twitter2017.

6.3.2 Interpretation with regards to the modality merging

Each multimodal model has differing approaches to merging the language and image. From the results showing differences for image versus not, we can again look at the zero-shot columns in Table 5.9 and Table 5.7, and what is perhaps the most relative or noteworthy gain there is from the OpenFlamingo model, posting upwards of 5% gains in F1 score. But as we see, this is still volatile as we also have prompts causing decreased performance. And the fact that the OpenFlamingo model is the one that has the largest relative decreased performance relative to its underlying LLM.

Still, while there are many confounding factors, the zero-shot results on the Twitter datasets demonstrate that OpenFlamingo can perform relatively better with significant results (for zero-shot) by having visuals included. But there are too many circumstantial factors to say anything generalizable about the effectiveness of different approaches to modality merging.

6.3.3 The image modality data

From both our results and the surveyed data, we see that current methods are only able to leverage images for around a 1-2% increase in the F1 scores. As pointed out by Moon et al. [57], the experiment will show the ability of a method to leverage the imaging modality depending on the underlying data's potential. From the surveyed research and experiment, it is unclear what the highest relative performance for the underlying dataset is, and by virtue of there only being two openly accessible MNER datasets, which also only have data from one domain, the current research does not have great generalizable demonstrative power. Meaning that any generalizable relative effect of including images for named entity recognition is hard to derive from the currently accessible datasets or potentially any restricted sets since the potential is always dependent on the underlying data.

Chapter 7

Conclusions and future work

In this chapter we review our work in light of the results and discussion, conclude our findings, and propose further work.

7.1 Conclusion

Our work involved building pipelines with various LLMs and MLLMs that could modify the standard named entity recognition task to apply to generative pre-trained models. Then we adopted a prompt engineering and a few-shot in-context learning approach, where we varied the prompts and number of examples fed to the models.

Three research questions guided our work:

Research question 1 *Can prompting generative LLMs be a viable alternative approach for named entity recognition?*

Research question 2 *Is performance increased when integrating visuals for named entity recognition via pre-trained multimodal models?*

Research question 3 *Does more examples improve performance for the LLMs and multimodal models?*

Regarding $RQ1$. From our results and analysis, we find that some LLMs outperform the other competing approaches and models under few-shot circumstances, especially for zero-shot learning, also because the amount of existing approaches able to do zero-shot for NER is tiny. But also compared to surveyed low-resource results, our method performs well and outperforms the baseline method we constructed, and rank well against surveyed methods. Especially for zero-shot, where we improve drastically (10-20%) over the comparable methods depending on the dataset. It then seems that with the right pre-trained model, the answer to $RQ1$ is yes, for a low-resource environment and given the right pre-trained model.

For $RQ2$ we find that under some conditions, circumstantial based on the prompt, examples, and data. There is a slight performance increase when integrating the visuals. And that there are as many circumstances where it does not help. The relative gain is small at around a few percentage points, which is consistent with existing literature examining integrating images with NER. On the other hand, while existing research has attempted specialized approaches for constructing models able to leverage the visual component, our work finds that the off-the-shelf visual-language models are able to do this for NER without much work (in our case all we provided was an instructive sentence saying the image was supposed to help). But, that the gains only come circumstantially, the method can also show relatively decreased performance for similar conditions and are as such quite volatile with regards to the prompt. But we also find that the existing literature has a lacking analysis of relative gains, with most only looking to report their top-scoring performance and not doing ablations with regards to image versus no-image. This also makes the feasible gain from the data hard to reflect on.

$RQ3$ has a more diffuse conclusion in our case. The relative gain that comes with a few training examples also varies for each model, prompt, and data. There are many cases it is true that more examples improve performance. But the underlying data poses us questions regarding our methods. Some models have relatively consistent gains with prompts, while those, especially those that have the strongest zero-shot performance start having lower or stable F1 scores, but with the inclusion of a low-occurring MISC entity type start performing better at that specific entity.

We see when examining the MISC entity type that it poses some problems, which we think are more novel for QA-NER and our methods than earlier research. Our approach seems to benefit from the entity classes having fairly understandable names to leverage the pre-trained model. From the data, we expect that for NER tasks utilizing the "other" class for collecting the entities not enumerated by the ones provided, our method can have decreased performance, due to it not being able to cope very well with the identification of entities other than the types provided which has a direct understandable class to them. For circumstances where there is no need for a MISC type, we expect our approach with those Flan-T5 models to fare better than in the alternative, which we see is the case for the MIT Restaurant and MIT Movies datasets.

Finally, we conclude that our work has brought results forward that are in line with the movement of leveraging large generalizable models for new tasks [79], and our work has provided non-trivial results for the ability of LLMs and MLLMs to do named entity recognition. Our work also prompts some questions that we try to synthesize into future work in section 7.2.

7.2 Future work

In this section, we provide some suggestions and thoughts regarding future work. We attempt to split these topics into data, approaches, and models.

7.2.1 Datasets

The underlying data that the method is benchmarked against is obviously very important. We propose some different data that could be used in a similar experiment.

Alternatively structured NER: Our method, as it is applicable to sparse data, could be tested for a large range of NER data. Notably, data that shows greater granularity, and is nested, or has overlapping disjoint entity types. Both should not be very difficult to adapt to, possibly could it be easier than flat-NER since the asking of N-entities can handle nested entities without having to flatten the conflicting entity types. The main issue with more granular entity types is that, for our approach, the run-time across the data increases linearly with the number of entity types. Which for very large models could introduce a significant computational cost.

Other available datasets: A dataset we considered but was out of scope given the included work was CrossNER [52] which handles several domains. Our experiment only handled data from the news, restaurant, movies and Twitter domain. Another notable dataset is FewNERD by Ding et al. [22], which is explicitly made for few-shot NER. One factor that made us not go with this dataset is that the creators intended for the benchmarking that the models are tuned on some training data sectioned according to entity types, and then have some query data with the relevant entity types for fine-tuning. The proposed benchmark would in our case then seem to be only inference over some fine classes belonging to two coarse-grained entity types which correspond to location and organization [22].

7.2.2 Alternative approaches

Our approach focused only on a singular example of what can be done with LLMs and MLLMs. We suggest some alternatives.

Modalities and data handling

There are other classes of modalities that could be worth looking into. Flamingo by example, also handles **video** content, so there may be some cases where video and text go together, either in social media or on short-video content sites which could benefit.

Direct **audio** is also sometimes available together with textual description, meaning that you could have transcription from an audio source, which then should be tagged. There may be latent information in the audio data that could add to the ability to understand what entities are in the text.

Interleaved images and text

Our work used images only from the example input at inference, not from the training examples. As more models are able to deal with interleaved images and text, it might be interesting to see if the multimodal models are able to leverage in-context examples of images to further understand, or align, the visual component with the instructive language task.

Prompting

Different approaches to the prompts are an area that could improve performance. Our prompts have had a very limited design process. Chain of thought (COT) prompting is an exciting domain of prompting. Although we have not seen a way to do a chain of thought zero-shot prompting, it can improve upon the manually made prompt templates in a few-shot environment. A rephrasing of the prompt to be not in the question answering form, but in some sort of filling out of an empty placeholder is also possible.

Fine tuning: Since our experiments consisted of only in-context learning, we would be interested in seeing how they compare vs fine-tuning with very few examples. The obvious downside is that for such large models, the computational cost of fine-tuning with large amounts of differing amounts of prompts and training examples can be burdensome, but can be worth looking into.

Prompt tuning: Prompt tuning [43] could be an efficient way to avoid the volatile prompt engineering process, and rather tune the prompt to most effectively increase performance. We consider this interesting both for the unimodal case, and perhaps especially for the multimodal case, as we saw that the relative gain, or loss, from visuals, was surprisingly volatile with regard to the prompt.

7.2.3 Models

The models, especially the multimodal models we have used have been very much a result of what is currently openly available. As we see more models being created and made publicly available, there are some existing models which would be of interest for similar experiments, and some other modality fusion.

Flamingo: Flamingo [1] has been the benchmark comparison for much of our multimodal models [5]. While there have been larger generalizable visual-language models released recently, Flamingo has an explicit few-shot learning focus, making this an interesting model which we think has potential for further experiment. The development of OpenFlamingo [108] also continues, so it's possible more interesting checkpoints of that will be released.

InstructBLIP: As a further development of BLIP2, InstructBLIP [18] was newly released. It builds on BLIP2 but is instruction tuned as a multimodal model.

KOSMOS-1: The KOSMOS-1 model is a generalizable multimodal model by Microsoft, introduced by Huang et al. [34]. It has the ability to learn in-context, and like Flamingo can handle a wide range of tasks related to understanding visuals and language.

GPT-3 and GPT-4: A commonality for many of the multimodal models is that they mention or compare themselves to GPT-4 [61]. Our approach for LLMs should be adaptable to GPT-3 [7], as well as GPT-4. As these are very large performant models accessible through API, it should be possible to conduct a similar experiment to ours with them if someone sees it worthwhile relative to the costs.

Acknowledgements

I like to acknowledge the computational resources provided by NTNU via the IDUN cluster, without which I would not have been able to work with these large models [75].

Bibliography

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. URL `https://arxiv.org/abs/2204.14198`. [Last Accessed: 2023-06-24].

[2] Mohammed Albared, Marc Gallofré Ocaña, Abdullah S. Ghareb, and Tareq Al-Moslmi. Recent progress of named entity recognition over the most popular datasets. *2019 First International Conference of Intelligent Computing and Engineering (ICOICE)*, pages 1–9, 2019.

[3] Wikimedia Commons Aphex34. Fully connected convolutional neural network, 2015. URL `https://upload.wikimedia.org/wikipedia/commons/6/63/Typical_cnn.png`. [Last Accessed: 2023-06-24] This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) License. To view a copy of this license, visit `https://creativecommons.org/licenses/by-nc-sa/4.0/`.

[4] Simran Arora, Avanika Narayan, Mayee F. Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Re. Ask me anything: A simple strategy for prompting language models, 2022. URL `https://arxiv.org/abs/2210.02441`. [Last Accessed: 2023-06-24].

[5] Anas Awadalla and Irena Gao. Announcing openflamingo: An open-source framework for training vision-language models with in-context learning,

Mar 2023. URL https://laion.ai/blog/open-flamingo/. [Last Accessed: 2023-06-24].

[6] Anas Awadalla, Irena Gao, Joshua Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Open flamingo github repository, March 2023. URL https://doi.org/10.5281/zenodo.7733589.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[8] Sebastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023. URL https://arxiv.org/abs/2303.12712. Last Accessed: 2023-06-23.

[9] Maria Carmela Cariello, Alessandro Lenci, and Ruslan Mitkov. A comparison between named entity recognition models in the biomedical domain. In *Proceedings of the Translation and Interpreting Technology Online Conference*, pages 76–84, Held Online, July 2021. INCOMA Ltd. URL https://aclanthology.org/2021.triton-1.9. Last Accessed: 2023-06-13.

[10] Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pretraining. *Machine Intelligence Research*, 20:38–56, 01 2023. doi: 10.1007/s11633-022-1369-5.

[11] Shuguang Chen, Gustavo Aguilar, Leonardo Neves, and Thamar Solorio. Can images help recognize entities? a study of the role of images for multimodal NER. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 87–96, Online, November 2021. As-

sociation for Computational Linguistics. doi: 10.18653/v1/2021.wnut-1.11. URL https://aclanthology.org/2021.wnut-1.11. Last Accessed: 2023-06-13.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daume III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html. Last Accessed: 2023-06-13.

[13] Xiang Chen, Lei Li, Shumin Deng, Chuanqi Tan, Changliang Xu, Fei Huang, Luo Si, Huajun Chen, and Ningyu Zhang. LightNER: A lightweight tuning paradigm for low-resource NER via pluggable prompting. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2374–2387, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.209. Last Accessed: 2023-06-22.

[14] Xiang Chen, Ningyu Zhang, Lei Li, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. Good visual guidance make a better extractor: Hierarchical visual prefix for multimodal entity and relation extraction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1607–1618, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.121. URL https://aclanthology.org/2022.findings-naacl.121.

[15] Yanru Chen, Yanan Zheng, and Zhilin Yang. Prompt-based metric learning for few-shot ner, 2022. URL https://arxiv.org/abs/2211.04337. Last Accessed: 2023-06-21.

[16] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. https://arxiv.org/abs/2210.11416.

[17] Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.161. URL `https://aclanthology.org/2021.findings-acl.161`.

[18] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. `https://arxiv.org/abs/2305.06500`.

[19] Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. CONTaiNER: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.439. URL `https://aclanthology.org/2022.acl-long.439`.

[20] Luca Deininger, Bernhard Stimpel, Anil Yuce, Samaneh Abbasi-Sureshjani, Simon SchÃ¶nenberger, Paolo Ocampo, Konstanty Korski, and Fabien Gaire. A comparative study between vision transformers and cnns in digital pathology, 2022. `https://arxiv.org/abs/2206.00389`.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL `https://arxiv.org/abs/1810.04805`.

[22] Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.248. URL `https://aclanthology.org/2021.acl-long.248`.

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. URL `https://arxiv.org/abs/2010.11929`.

[24] Li Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4): 594–611, 2006. doi: 10.1109/TPAMI.2006.79.

[25] Jenny Rose Finkel and Christopher D. Manning. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Singapore, August 2009. Association for Computational Linguistics. URL `https://aclanthology.org/D09-1015`.

[26] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.295. URL `https://aclanthology.org/2021.acl-long.295`.

[27] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, and R. Feris. Spottune: Transfer learning through adaptive fine-tuning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4800–4809, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society. doi: 10.1109/CVPR.2019.00494. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00494`.

[28] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji-Rong Wen, and Jun Zhu. Pre-trained models: Past, present and future. *AI Open*, 2, 08 2021. doi: 10.1016/j.aiopen.2021.08.002.

[29] Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation, 2022. URL `https://arxiv.org/abs/2212.09611`.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL `https://arxiv.org/abs/1512.03385`.

[31] Hecht-Nielsen. Theory of the backpropagation neural network. In *International 1989 Joint Conference on Neural Networks*, pages 593–605 vol.1, 1989. doi: 10.1109/IJCNN.1989.118638.

[32] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*,

79(8):2554–2558, 1982. doi: 10.1073/pnas.79.8.2554. URL https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554.

[33] Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. Few-shot named entity recognition: An empirical baseline study. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10408–10423, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.813. URL https://aclanthology.org/2021.emnlp-main.813.

[34] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei. Language is not all you need: Aligning perception with language models, 2023. https://arxiv.org/abs/2302.14045.

[35] Yucheng Huang, Kai He, Yige Wang, Xianli Zhang, Tieliang Gong, Rui Mao, and Chen Li. COPNER: Contrastive learning with prompt guiding for few-shot named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2515–2527, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.222.

[36] Rahib Imamguluyev. The rise of gpt-3: Implications for natural language processing and beyond. *International Journal of Research Publication and Reviews*, 4:4893–4903, 03 2023. doi: 10.55248/gengpi.2023.4.33987.

[37] Meihuizi Jia, Lei Shen, Xin Shen, Lejian Liao, Meng Chen, Xiaodong He, Zhendong Chen, and Jiaqi Li. Mner-qg: An end-to-end mrc framework for multimodal named entity recognition with query grounding, 2022. URL https://arxiv.org/abs/2211.14739. Last accessed 25.06.2023.

[38] Meihuizi Jia, Xin Shen, Lei Shen, Jinhui Pang, Lejian Liao, Yang Song, Meng Chen, and Xiaodong He. Query prior matters: A mrc framework for multimodal named entity recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 3549â3558, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3548427. URL https://doi.org/10.1145/3503161.3548427.

[39] Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding language models to images for multimodal generation, 2023. URL https://arxiv.org/abs/2301.13823. Last accessed 16.06.2023.

[40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60 (6):84â90, may 2017. ISSN 0001-0782. doi: 10.1145/3065386. URL https://doi.org/10.1145/3065386.

[41] Qiuxia Lai, Salman Khan, Yongwei Nie, Jianbing Shen, Hanqiu Sun, and Ling Shao. Understanding more about human and machine attention in deep neural networks, 2019. URL https://arxiv.org/abs/1906.08764.

[42] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541.

[43] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243.

[44] Fei Li, ZhiChao Lin, Meishan Zhang, and Donghong Ji. A span-based model for joint overlapped and discontinuous named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4814–4828, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.372. URL https://aclanthology.org/2021.acl-long.372.

[45] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. URL https://arxiv.org/abs/2301.12597.

[46] Pei Li and Xinde Li. Multimodal fusion with co-attention mechanism. pages 1–8, 07 2020. doi: 10.23919/FUSION45008.2020.9190483. doi = 10.23919/FUSION45008.2020.9190483, https://ieeexplore.ieee.org/document/9190483.

[47] Andy T. Liu, Wei Xiao, Henghui Zhu, Dejiao Zhang, Shang-Wen Li, and Andrew Arnold. Qaner: Prompting question answering models for few-shot named entity recognition, 2022. URL `https://arxiv.org/abs/2203.01543`.

[48] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021. URL `https://arxiv.org/abs/2107.13586`.

[49] Shuheng Liu and Alan Ritter. Do conll-2003 named entity taggers still work well in 2023?, 2022. `https://arxiv.org/abs/2212.09747`.

[50] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.8. URL `https://aclanthology.org/2022.acl-short.8`.

[51] Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. Summary of chatgpt/gpt-4 research and perspective towards the future of large language models, 2023. `https://arxiv.org/abs/2304.01852`.

[52] Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. Crossner: Evaluating cross-domain named entity recognition, 2020. `https://arxiv.org/abs/2012.04373`.

[53] Di Lu, Leonardo Neves, Vitor Carvalho, Ning Zhang, and Heng Ji. Visual attention model for name tagging in multimodal social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1990–1999, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1185. URL `https://aclanthology.org/P18-1185`.

[54] Bhanuka Mahanama, Yasith Jayawardana, Sundararaman Rengarajan, Gavindya Jayawardena, Leanne Chukoskie, Joseph Snider, and Sampath Jayarathna. Eye movement and pupil measures: A review. *Frontiers in Computer Science*, 3, 2022. ISSN 2624-9898. doi: 10.3389/fcomp.

2021.733531. URL https://www.frontiersin.org/articles/10.3389/fcomp.2021.733531.

[55] Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

[56] Dan Milmo. Chatgpt reaches 100 million users two months after launch. *The Guardian*, Feb 2023. URL https://www.theguardian.com/technology/2023/feb/02/chatgpt-100-million-users-open-ai-fastest-growing-app.

[57] Seungwhan Moon, Leonardo Neves, and Vitor Carvalho. Multimodal named entity recognition for short social media posts. 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 852–860, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1078. URL https://aclanthology.org/N18-1078.

[58] Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines, 2021. https://arxiv.org/abs/2006.04884.

[59] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 689â696, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

[60] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning, 2018. URL https://arxiv.org/abs/1811.03378.

[61] OpenAI. Gpt-4 technical report, 2023. https://arxiv.org/abs/2303.08774.

[62] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks, 2015. URL https://arxiv.org/abs/1511.08458.

[63] Xu Paiheng. Qa-ner. https://github.com/paihengxu/QA-NER/tree/main, 2023. GitHub repository.

[64] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[65] Jun Qi, Jun Du, Sabato Marco Siniscalchi, Xiaoli Ma, and Chin-Hui Lee. On mean absolute error for deep neural network based vector-to-vector regression. *IEEE Signal Processing Letters*, 27:1485–1489, 2020. doi: 10.1109/lsp.2020.3016837. URL https://doi.org/10.1109%2Flsp.2020.3016837.

[66] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

[67] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[68] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020. ISSN 1532-4435.

[69] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124.

[70] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442.

[71] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. ISSN 00034851. URL http://www.jstor.org/stable/2236626.

[72] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2016. URL https://arxiv.org/abs/1609.04747.

[73] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986. https://arxiv.org/abs/1912.05848.

[74] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. 2003. doi: 10.48550/ARXIV.CS/0306050. URL https://arxiv.org/abs/cs/0306050.

[75] Magnus Själander, Magnus Jahre, Gunnar Tufte, and Nico Reissmann. EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure, 2019.

[76] Lin Sun, Jiquan Wang, Yindu Su, Fangsheng Weng, Yuxuan Sun, Zengwei Zheng, and Yuanyi Chen. RIVA: A pre-trained tweet multimodal model based on text-image relation for multimodal NER. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1852–1862, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.168. URL https://aclanthology.org/2020.coling-main.168.

[77] Lin Sun, Jiquan Wang, Kai Zhang, Yindu Su, and Fangsheng Weng. Rpbert: A text-image relation propagation-based bert model for multimodal ner, 2021. URL https://arxiv.org/abs/2102.02967. https://arxiv.org/abs/2102.02967.

[78] Lin Sun, Jiquan Wang, Kai Zhang, Yindu Su, and Fangsheng Weng. Rpbert: A text-image relation propagation-based bert model for multimodal ner. *ArXiv*, abs/2102.02967, 2021.

[79] Tian-Xiang Sun, Xiang-Yang Liu, Xi-Peng Qiu, and Xuan-Jing Huang. Paradigm shift in natural language processing. *Machine Intelligence Research*, 19(3):169–183, may 2022. doi: 10.1007/s11633-022-1331-6. URL https://doi.org/10.1007%2Fs11633-022-1331-6.

[80] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothee Lacroix, Baptiste Roziere, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[81] Sowmya Vajjala and Ramya Balasubramaniam. What do we really know about state of the art ner?, 2022. https://arxiv.org/abs/2205.00034.

[82] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762. [Last Accessed: 2023-06-24].

[83] Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Åukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. Tensor2tensor for neural machine translation. 03 2018. URL https://www.researchgate.net/figure/The-Transformer-model-architecture_fig1_323904682. [Last Accessed: 2023-06-24] This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) License. To view a copy of this license, visit https://creativecommons.org/licenses/by-nc-sa/4.0/.

[84] Xinyu Wang, Min Gui, Yong Jiang, Zixia Jia, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. Ita: Image-text alignments for multi-modal named entity recognition, 2021. URL https://arxiv.org/abs/2112.06482.

[85] Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. Automated concatenation of embeddings for structured prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2643–2660, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.206. URL https://aclanthology.org/2021.acl-long.206.

[86] Xuwu Wang, Junfeng Tian, Min Gui, Zhixu Li, Jiabo Ye, Ming Yan, and Yanghua Xiao. Promptmner: Prompt-based entity-related visual clue extraction andÂ integration forÂ multimodal named entity recognition. In Arnab Bhattacharya, Janice Lee Mong Li, Divyakant Agrawal, P. Krishna Reddy, Mukesh Mohania, Anirban Mondal, Vikram Goyal, and Rage

Uday Kiran, editors, *Database Systems for Advanced Applications*, pages 297–305, Cham, 2022. Springer International Publishing. ISBN 978-3-031-00129-1.

[87] Zilong Wang and Jingbo Shang. Towards few-shot entity recognition in document images: A label-aware sequence-to-sequence framework. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4174–4186, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.329. URL https://aclanthology.org/2022.findings-acl.329.

[88] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022. https://arxiv.org/abs/2109.01652.

[89] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. https://arxiv.org/abs/2201.11903.

[90] Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. Larger language models do in-context learning differently, 2023. https://arxiv.org/abs/2303.03846.

[91] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. A prompt pattern catalog to enhance prompt engineering with chatgpt, 2023. https://arxiv.org/abs/2302.11382.

[92] Wikipedia. Precision and recall — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Precision%20and%20recall&oldid=1122267443, 2022. [Online; accessed 13-December-2022].

[93] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for

Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

[94] Bo Xu, Shizhou Huang, Chaofeng Sha, and Hongya Wang. Maf: A general matching and alignment framework for multimodal named entity recognition. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 1215â1223, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391320. doi: 10.1145/3488560.3498475. URL https://doi.org/10.1145/3488560.3498475.

[95] Zhiyang Xu, Ying Shen, and Lifu Huang. Multiinstruct: Improving multimodal zero-shot learning via instruction tuning, 2022. https://arxiv.org/abs/2212.10773.

[96] Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1182.

[97] Jie Yang, Soyeon Caren Han, and Josiah Poon. A survey on extraction of causal relations from natural language text, 2021. URL https://arxiv.org/abs/2101.06426.

[98] Yi Yang and Arzoo Katiyar. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.516. URL https://aclanthology.org/2020.emnlp-main.516.

[99] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *AAAI Conference on Artificial Intelligence*, 2021.

[100] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chaoya Jiang, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. mplug-owl: Modularization empowers large language models with multimodality, 2023.

[101] Seonghyeon Ye, Hyeonbin Hwang, Sohee Yang, Hyeongu Yun, Yireun Kim, and Minjoon Seo. In-context instruction learning, 2023. https://arxiv.org/abs/2302.14691.

[102] Jianfei Yu, Jing Jiang, Li Yang, and Rui Xia. Improving multimodal named entity recognition via entity span detection with unified multimodal transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3342–3352, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.306. URL `https://aclanthology.org/2020.acl-main.306`.

[103] Dong Zhang, Suzhong Wei, Shoushan Li, Hanqian Wu, Qiaoming Zhu, and Guodong Zhou. Multi-modal graph fusion for named entity recognition with targeted visual guidance. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14347–14355, May 2021. doi: 10.1609/aaai.v35i16.17687. URL `https://ojs.aaai.org/index.php/AAAI/article/view/17687`.

[104] Dong Zhang, Suzhong Wei, Shoushan Li, Hanqian Wu, Qiaoming Zhu, and Guodong Zhou. Multi-modal graph fusion for named entity recognition with targeted visual guidance. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14347–14355, May 2021. doi: 10.1609/aaai.v35i16.17687. URL `https://ojs.aaai.org/index.php/AAAI/article/view/17687`.

[105] Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. Adaptive co-attention network for named entity recognition in tweets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11962. URL `https://ojs.aaai.org/index.php/AAAI/article/view/11962`.

[106] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. `https://arxiv.org/abs/2205.01068`.

[107] Fei Zhao, Chunhui Li, Zhen Wu, Shangyu Xing, and Xinyu Dai. Learning from different text-image pairs: A relation-enhanced graph convolutional network for multimodal ner. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 3983â3992, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3548228. URL `https://doi.org/10.1145/3503161.3548228`.

[108] Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang,

and Yejin Choi. Multimodal c4: An open, billion-scale corpus of images interleaved with text, 2023. https://arxiv.org/abs/2304.06939.

[109] Morteza Ziyadi, Yuting Sun, Abhishek Goswami, Jade Huang, and Weizhu Chen. Example-based named entity recognition, 2020. https://arxiv.org/abs/2008.10570.

Appendices

A Inference parameters

Parameter	**Value**	**Comment**
Temperature	0	Set to 0 for less randomness.
Inference batch size	1	Some of the models acted unstable with batches higher than 1, so for the final results presented in this thesis, batch inference was run at 1. During development we used batching for those models that were stable, like Flan-related models.
Search parameter	OpenFlamingo	Beam search with beam size of 3 (as in the original Flamingo paper)
	Others	Greedy decoding/Greedy search
Max new tokens	20	Set to 20 to limit the lenght of answers that were likely to be generated our of the models.
Max input lenght	>1024	All models, except for BERT was configured to be able to handle inputs of at least 1024 tokens without truncation.

Table A1: Relevant inference parameters for the experiment.

B Model repositories and weights

Model name	**Github repo (github.com/+)**	**Comment**
FROMAGe	kohjingyu/fromage	Used the standard checkpoint, not the additionally stronger linear layer.
OpenFlamingo	mlfoundations/open_flamingo	Needs self-sourced LLAMA weights.
BLIP2	salesforce/LAVIS	Huggingface was used instead of original repo.
mPLUG-Owl	X-PLUG/mPLUG-Owl	Needs self-sourced LLAMA weights.

Table B2: Github repositories that were used for the models. Other models than the ones mentioned here were integrated in the transformers/huggingface library.

Model group	**Model name**	**Huggingface weight link (huggingface.co/)**	**Comment**
Language models	Flan-T5-XL	google/flan-t5-xl	
	Flan-T5-XXL	google/flan-t5-xxl	
	OPT-2.7b	facebook/opt-2.7b	
	OPT-6.7b	facebook/opt-6.7b	
	LLAMA	decapoda-research/llama-7b-hf	We got the licence for LLAMA weights (required for usage) and used these huggingface weights for practicality later
	QA-NER	deepset/bert-large-uncased-whole-word-masking-squad2	Weights for the pre-trained BERT model
BLIP2	Flan-T5-XL	Salesforce/blip2-flan-t5-xl	
	Flan-T5-XXL	Salesforce/blip2-flan-t5-xxl	
	OPT-2.7b	Salesforce/blip2-opt-2.7b	
	OPT-6.7b	Salesforce/blip2-opt-6.7b	
mPLUG-Owl		MAGAer13/mplug-owl-llama-7b-ft	
OpenFlamingo		openflamingo/OpenFlamingo-9B	
FROMAGe		N/A	Weights are included in the FROMAGe repo

Table B3: Almost all weights for the models in the experiment were sourced from huggingface, except for FROMAGe. Models not totally integrated into huggingface still published their weights there.

C Software

Software/library	**Comment**
Pandas	Data analysis library used for handling datasets [55].
PyTorch	Machine learning framework [64]
Transformers	Library from Huggingface integrating with PyTorch for handling of models and weights [93]
Python	We used python 3.9.12 for all experiments

Table C4: Notable software used for the experiment.

D Additional experimental results

Twitter2015	**With image**									
Model	**Prompt**	**Zero-shot**	**1-shot**	**2-shot**	**3-shot**	**4-shot/ 1-set**	**5-shot**	**6-shot**	**7-shot**	**8-shot/ 2-sets**
FROMAGe	1	0.00	0.09	0.12	0.14	0.13	0.14	0.14	0.15	**0.17**
	2	0.01	0.09	0.14	0.15	0.15	0.16	0.15	0.15	**0.17**
	3	0.01	0.10	0.13	0.15	0.14	**0.16**	0.15	0.14	0.16
	4	0.01	0.13	0.14	0.16	0.15	0.16	0.18	0.18	**0.19**
	5	0.02	0.12	0.16	0.17	0.16	0.17	0.18	0.18	**0.19**
BLIP2-opt-2.7b	1	0.01	0.07	0.11	0.11	0.12	0.14	**0.16**	0.15	0.14
	2	0.01	0.06	0.09	0.11	0.11	0.15	0.17	0.16	**0.18**
	3	0.01	0.05	0.09	0.12	0.13	0.14	**0.17**	0.17	0.17
	4	0.03	0.08	0.12	0.13	0.12	0.15	0.16	**0.18**	0.18
	5	0.03	0.08	0.11	0.13	0.13	0.15	0.16	0.17	**0.18**
BLIP2-opt-6.7b	1	0.01	0.10	0.15	0.17	0.19	0.20	**0.22**	0.21	0.20
	2	0.00	0.11	0.12	0.17	**0.19**	0.18	0.19	0.17	0.17
	3	0.00	0.11	0.14	0.17	**0.18**	0.18	0.18	0.16	0.17
	4	0.02	0.14	0.18	0.19	0.20	0.19	0.21	**0.20**	0.19
	5	0.03	0.14	0.17	0.20	0.20	0.20	**0.21**	0.21	0.20
BLIP2-flan-t5-xl	1	**0.60**	0.57	0.55	0.53	0.54	0.55	0.55	0.52	0.53
	2	0.55	**0.60**	0.60	0.58	0.58	0.58	0.58	0.54	0.55
	3	**0.61**	0.60	0.59	0.56	0.57	0.57	0.57	0.53	0.54
	4	**0.64**	0.60	0.59	0.56	0.56	0.57	0.57	0.54	0.54
	5	0.50	**0.59**	0.59	0.58	0.59	0.59	0.58	0.55	0.56
BLIP2-flan-t5-xxl	1	0.40	0.40	0.38	0.35	0.38	0.39	**0.41**	0.39	0.40
	2	0.59	**0.60**	0.56	0.52	0.52	0.52	0.52	0.49	0.50
	3	**0.52**	0.51	0.50	0.49	0.50	0.50	0.50	0.50	0.50
	4	**0.50**	0.48	0.49	0.49	0.50	0.50	0.50	0.50	0.49
	5	0.51	**0.52**	0.49	0.49	0.50	0.50	0.50	0.50	0.50
mPLUG-Owl	1	0.30	0.22	0.29	0.32	0.32	**0.33**	0.32	0.31	0.30
	2	0.29	0.33	0.31	0.33	0.33	**0.34**	0.34	0.33	0.31
	3	0.26	**0.31**	0.29	0.31	0.31	0.31	0.31	0.30	0.30
	4	**0.36**	0.30	0.33	0.34	0.34	0.36	0.36	0.34	0.34
	5	**0.37**	0.35	0.35	0.35	0.37	0.37	0.37	0.35	0.34
OpenFlamingo	1	0.13	0.20	0.28	0.32	0.31	**0.34**	0.33	0.30	0.29
	2	0.15	0.22	0.24	0.25	**0.27**	0.27	0.24	0.20	0.19
	3	0.14	0.21	0.25	0.25	0.26	**0.27**	0.24	0.20	0.19
	4	0.13	0.25	0.32	0.33	0.33	**0.34**	0.32	0.28	0.26
	5	0.11	0.27	0.32	0.34	0.34	0.35	**0.36**	0.32	0.30

Table D5: Mean Micro-F1 scores for the Twitter2015 dataset using the stringent evaluation metric when using the datasets images. Score averages are for models with their corresponding prompts and amount of random N-singular few-shot examples. The highest scores for each row are bolded with a preference for smaller amounts of training data, the best score for each column is underlined.

Twitter2017	**With image**									
Model	**Prompt**	**Zero-shot**	**1-shot**	**2-shot**	**3-shot**	**4-shot/ 1-set**	**5-shot**	**6-shot**	**7-shot**	**8-shot/ 2-sets**
FROMAGe	1	0.00	0.09	0.15	0.19	0.18	0.20	0.20	**0.21**	0.17
	2	0.01	0.08	0.13	**0.19**	0.19	0.18	0.19	0.17	0.19
	3	0.01	0.08	0.12	**0.19**	0.18	0.19	0.19	0.18	0.18
	4	0.00	0.13	0.16	0.21	0.20	**0.22**	0.21	0.20	0.20
	5	0.02	0.13	0.14	0.20	0.20	0.19	0.22	**0.23**	0.23
BLIP2-opt-2.7b	1	0.01	0.11	0.07	0.10	**0.17**	0.15	0.15	0.15	0.13
	2	0.01	0.07	0.04	0.05	0.09	0.14	0.12	0.13	**0.18**
	3	0.00	0.07	0.04	0.05	0.09	0.14	0.13	0.14	**0.17**
	4	0.03	0.13	0.06	0.10	0.15	0.15	0.14	**0.16**	0.15
	5	0.04	0.10	0.05	0.08	0.15	0.13	0.15	**0.18**	0.14
BLIP2-opt-6.7b	1	0.00	0.15	0.21	0.22	0.23	0.19	0.17	0.25	**0.26**
	2	0.00	0.16	0.18	0.22	0.23	0.22	0.20	0.25	**0.27**
	3	0.00	0.15	0.18	0.22	0.23	0.22	0.20	0.24	**0.27**
	4	0.02	0.19	0.23	0.25	0.26	0.25	0.22	0.27	**0.28**
	5	0.02	0.21	0.22	0.24	0.27	0.26	0.23	0.27	**0.29**
BLIP2-flan-t5-xl	1	0.55	0.55	**0.56**	0.56	0.56	0.56	0.56	0.56	0.56
	2	0.46	0.51	0.51	0.51	0.51	**0.52**	0.52	0.51	0.51
	3	**0.55**	0.55	0.55	0.54	0.54	0.54	0.55	0.54	0.54
	4	**0.57**	0.57	0.55	0.55	0.55	0.55	0.55	0.55	0.55
	5	0.42	**0.52**	0.51	0.51	0.51	0.51	0.51	0.51	0.51
BLIP2-flan-t5-xxl	1	0.41	0.37	**0.44**	0.44	0.43	0.44	0.40	0.41	0.41
	2	**0.57**	0.54	0.53	0.53	0.52	0.52	0.52	0.52	0.52
	3	**0.52**	0.50	0.53	0.52	0.52	0.52	0.51	0.52	0.51
	4	0.48	0.49	**0.52**	0.51	0.51	0.51	0.51	0.51	0.51
	5	**0.50**	0.48	0.50	0.50	0.50	0.50	0.50	0.50	0.50
mPLUG-Owl	1	0.29	0.29	0.30	0.31	**0.32**	0.27	0.27	0.29	0.29
	2	0.27	**0.28**	0.24	0.27	0.28	0.24	0.26	0.27	0.27
	3	0.25	**0.28**	0.24	0.26	0.27	0.23	0.24	0.25	0.26
	4	**0.34**	0.32	0.29	0.32	0.33	0.28	0.31	0.33	0.33
	5	0.32	0.30	0.29	0.33	**0.34**	0.29	0.32	0.34	0.34
OpenFlamingo	1	0.13	0.31	0.26	0.31	0.31	0.33	**0.32**	0.31	0.31
	2	0.13	0.18	0.13	0.19	0.24	0.27	**0.28**	0.25	0.25
	3	0.12	0.20	0.16	0.22	0.27	**0.28**	0.28	0.26	0.27
	4	0.13	**0.33**	0.27	0.31	0.33	0.33	0.33	0.32	0.33
	5	0.11	0.32	0.23	0.30	0.31	**0.34**	0.33	0.31	0.31

Table D6: Mean Micro-F1 scores for the Twitter2017 dataset using the stringent evaluation metric when using the dataset's images. Score averages are for models with their corresponding prompts and amount of random N-singular few-shot examples. Highest scores for each row is bolded with preference for smaller amounts of training data, best score for each column is underlined.

Model	**Prompt**	**Zero-shot**	**1-shot**	**2-shot**	**3-shot**	**4-shot/ 1-set**	**5-shot**	**6-shot**	**7-shot**	**8-shot/ 2-sets**
opt-2.7b	1	0.03	0.16	0.17	0.18	0.18	0.18	**0.20**	0.20	0.19
	2	0.03	0.11	0.15	0.16	0.18	0.17	0.19	**0.20**	0.19
	3	0.02	0.12	0.16	0.17	0.19	0.18	**0.19**	0.19	0.19
	4	0.03	0.16	0.19	0.21	0.21	0.20	**0.22**	0.22	0.22
	5	0.03	0.15	0.17	0.18	0.19	0.19	0.21	**0.22**	0.22
opt-6.7b	1	0.00	0.18	0.20	0.21	**0.23**	0.22	0.23	0.22	0.21
	2	0.00	0.16	0.18	0.20	0.22	0.23	0.22	**0.24**	0.23
	3	0.00	0.13	0.17	0.20	0.21	0.21	**0.22**	0.22	0.22
	4	0.00	0.18	0.20	0.22	0.24	0.24	**0.25**	0.24	0.25
	5	0.00	0.17	0.21	0.22	0.24	0.24	0.24	**0.25**	0.24
flan-t5-xl	1	**0.57**	0.55	0.56	0.56	0.55	0.56	0.55	0.55	0.54
	2	**0.54**	0.52	0.51	0.50	0.50	0.50	0.50	0.50	0.51
	3	**0.59**	0.55	0.54	0.53	0.54	0.54	0.54	0.54	0.54
	4	**0.60**	0.56	0.55	0.54	0.54	0.53	0.53	0.53	0.53
	5	**0.53**	0.53	0.51	0.50	0.50	0.50	0.50	0.50	0.50
flan-t5-xxl	1	0.41	0.34	0.46	0.47	0.46	**0.48**	0.44	0.46	0.46
	2	**0.58**	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
	3	0.52	0.51	**0.53**	0.52	0.53	0.53	0.52	0.52	0.53
	4	0.49	0.52	**0.53**	0.52	0.52	0.52	0.52	0.52	0.53
	5	0.49	0.49	**0.51**	0.51	0.51	0.51	0.51	0.51	0.51
llama-7b	1	0.25	0.28	0.28	0.28	0.30	**0.30**	0.29	0.29	0.29
	2	0.21	0.27	0.27	0.28	0.29	0.29	0.29	0.29	**0.30**
	3	0.17	0.26	0.27	0.27	0.28	**0.29**	0.28	0.29	0.29
	4	0.27	**0.31**	0.31	0.30	0.31	0.31	0.30	0.31	0.30
	5	0.29	0.31	0.29	0.30	0.31	**0.32**	0.30	0.31	0.31

Table D7: Mean Micro-F1 scores for unimodal models on the Twitter2017 dataset using the stringent evaluation metric when not using the dataset's images. Score averages are for models with their corresponding prompts and amount of random N-singular few-shot examples. Highest scores for each row is bolded with preference for smaller amounts of training data, best score for each column is underlined.

Model	**Prompt**	**Zero-shot**	**1-shot**	**2-shot**	**3-shot**	**4-shot/ 1-set**	**5-shot**	**6-shot**	**7-shot**	**8-shot/ 2-sets**
opt-2.7b	1	0.02	0.12	0.12	0.15	0.15	0.17	**0.21**	0.20	0.18
	2	0.01	0.11	0.13	0.19	0.17	0.18	0.21	**0.23**	0.23
	3	0.01	0.09	0.14	0.18	0.16	0.18	**0.21**	0.21	0.21
	4	0.02	0.12	0.15	0.18	0.17	0.19	0.23	**0.24**	0.23
	5	0.02	0.11	0.15	0.18	0.18	0.20	0.23	**0.24**	0.24
opt-6.7b	1	0.00	0.10	0.12	0.18	**0.19**	0.17	0.19	0.19	0.18
	2	0.00	0.08	0.11	0.16	0.17	0.17	0.17	0.17	**0.18**
	3	0.00	0.08	0.11	0.15	0.16	0.15	0.16	0.16	**0.17**
	4	0.00	0.13	0.15	0.18	0.19	0.19	**0.21**	0.19	0.19
	5	0.00	0.11	0.13	0.20	0.21	0.21	**0.22**	0.20	0.20
flan-t5-xl	1	0.55	**0.58**	0.57	0.55	0.56	0.56	0.56	0.54	0.55
	2	**0.59**	0.59	0.59	0.57	0.57	0.57	0.57	0.53	0.53
	3	**0.63**	0.61	0.59	0.57	0.57	0.57	0.57	0.54	0.55
	4	**0.63**	0.59	0.58	0.56	0.56	0.57	0.57	0.54	0.54
	5	**0.59**	0.58	0.57	0.55	0.55	0.55	0.55	0.52	0.52
flan-t5-xxl	1	0.40	0.37	0.40	0.35	0.39	0.41	**0.43**	0.40	0.42
	2	0.57	**0.58**	0.54	0.51	0.52	0.52	0.52	0.48	0.49
	3	0.49	0.49	0.49	0.49	**0.50**	0.50	0.49	0.49	0.50
	4	0.48	0.49	0.49	0.49	**0.50**	0.50	0.50	0.50	0.50
	5	0.49	**0.50**	0.50	0.49	0.50	0.50	0.50	0.49	0.49
llama-7b	1	0.25	0.14	0.20	0.24	0.25	**0.27**	0.26	0.25	0.25
	2	0.20	0.17	0.18	0.23	0.23	0.25	**0.25**	0.24	0.25
	3	0.17	0.14	0.17	0.21	0.23	**0.25**	0.24	0.24	0.24
	4	0.28	0.20	0.25	0.26	0.28	**0.29**	0.29	0.27	0.28
	5	0.28	0.20	0.24	0.27	0.27	**0.29**	0.28	0.27	0.27

Table D8: Mean Micro-F1 scores for unimodal models on Twitter2015 dataset using the stringent evaluation metric when not using the dataset's images. Score averages are for models with their corresponding prompts and amount of random N-singular few-shot examples. Highest scores for each row is bolded with preference for smaller amounts of training data, best score for each column is underlined.

Model	Prompt	Zero-shot	1-shot	2-shot	3-shot	4-shot/ 1-set	5-shot	6-shot	7-shot	8-shot/ 2-sets
FROMAGe	1	0	0.05	0	0.02	0.01	0	-0.01	-0.01	-0.02
	2	0.01	0.03	-0.01	0.03	0.01	0	-0.02	-0.01	-0.02
	3	0	0.06	0.01	0.03	0.01	0	0	0	0
	4	0.01	0.04	0	0.03	0.02	0	-0.02	-0.02	-0.03
	5	0.02	0.02	-0.02	0.02	0.01	-0.01	-0.02	-0.02	-0.02
BLIP2-opt-2.7b	1	0	-0.01	-0.11	-0.06	-0.03	0	0.02	-0.01	-0.01
	2	-0.03	0.01	-0.13	-0.1	-0.07	-0.06	-0.01	-0.02	-0.02
	3	-0.05	0.01	-0.09	-0.05	-0.05	-0.02	0	-0.02	0
	4	-0.04	0.02	-0.1	-0.06	-0.02	0	0.01	0.01	0.01
	5	-0.01	0.03	-0.12	-0.06	-0.03	0	0.01	0.01	0.01
BLIP2-opt-6.7b	1	0	0.08	0.03	0.05	0.03	0.02	-0.02	-0.02	-0.04
	2	0	0.09	0.04	0.06	0.04	0.04	0.01	0.03	0
	3	0	0.13	0.07	0.07	0.04	0.03	0.03	0.02	0
	4	0	0.06	0.06	0.07	0.03	0.02	0	-0.01	-0.04
	5	0.01	0.06	0.04	0.06	0.04	0.02	0.01	-0.01	-0.04
BLIP2-flan-t5-xl	1	-0.03	0.01	0	0.01	-0.01	0	0	0.01	0
	2	-0.07	0.01	0.01	0.01	0	0	0	0	0
	3	-0.09	0.01	0.01	0.01	0.01	0	0.01	0	0
	4	-0.02	0.01	0.02	0	0.01	0	-0.01	0.01	0
	5	-0.12	0.01	-0.01	0	0	0	-0.01	0	0
BLIP2-flan-t5-xxl	1	0	0.01	0	0.01	0	0	0.01	0	0.01
	2	-0.01	0.01	0.01	0.01	0.01	0.01	0.01	0	0.01
	3	0.02	0.01	0.02	0.02	0.02	0.01	0.01	0.01	0.01
	4	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0	0.01
	5	0.01	0.02	0.01	0.01	0.01	0.01	0.02	0.01	0.01
mPLUG-Owl	1	0.04	0.01	0.01	0.02	0.02	0	0	-0.02	-0.01
	2	0.11	0.02	0.03	0.02	0.02	0.02	0.02	0	0.01
	3	0.06	0.02	0.03	0.02	0.02	0.01	0	-0.01	0.02
	4	0.01	0	-0.01	0.01	-0.01	-0.01	0.01	-0.01	0.01
	5	0.04	0	0.01	0	0	0	-0.01	0.01	0
OpenFlamingo	1	-0.14	-0.05	-0.09	-0.09	-0.05	-0.08	-0.08	-0.07	-0.07
	2	-0.1	-0.17	-0.15	-0.17	-0.14	-0.16	-0.14	-0.15	-0.14
	3	-0.09	-0.15	-0.17	-0.16	-0.15	-0.16	-0.16	-0.16	-0.15
	4	-0.15	-0.1	-0.11	-0.13	-0.09	-0.1	-0.1	-0.08	-0.08
	5	-0.18	-0.13	-0.12	-0.16	-0.12	-0.12	-0.13	-0.1	-0.11

Table D9: Relative difference between Micro-F1 scores for the CONLL dataset when subtracting the underlying LLM from the MLLM. Score averages are for models with their corresponding prompts and amount of random N-singular few-shot examples.

Model	Prompt	Zero-shot	1-shot	2-shot	3-shot	4-shot	5-shot	6-shot	7-shot	8-shot/ 1-set
FROMAGe	1	0.01	0	-0.02	0	-0.03	-0.03	-0.02	-0.01	-0.02
	2	0.01	0	-0.02	-0.02	-0.02	-0.02	-0.03	-0.01	-0.01
	3	0	0	-0.02	0	-0.01	-0.01	-0.03	-0.01	-0.02
	4	0.03	-0.01	-0.03	-0.02	-0.02	-0.04	-0.05	-0.04	-0.02
	5	0.03	-0.01	-0.04	-0.03	-0.03	-0.04	-0.05	-0.04	-0.03
BLIP2-opt-2.7b	1	0	0.03	-0.05	-0.05	-0.04	-0.01	0	0	-0.01
	2	0	0.02	-0.05	-0.06	-0.08	-0.04	-0.03	-0.02	-0.01
	3	-0.01	0.03	-0.06	-0.05	-0.06	-0.05	-0.04	-0.03	-0.02
	4	-0.01	0.02	-0.05	-0.06	-0.05	-0.02	-0.02	-0.01	-0.01
	5	0.01	0.03	-0.06	-0.06	-0.05	-0.05	-0.02	-0.01	0
BLIP2-opt-6.7b	1	0	0.02	0	0.02	0	0	0.01	0.01	-0.01
	2	0	0.03	0.01	0	0.01	0.01	0.01	0.02	0.01
	3	0	0.04	0.01	0.02	0.01	0.01	0	0.01	0
	4	0	0.03	0.01	0.01	0.02	0	-0.02	-0.01	0
	5	0	0.03	0.02	0.02	0.01	0	-0.01	0	-0.01
BLIP2-flan-t5-xl	1	0	0	0.01	0	0	-0.01	0	0.01	0
	2	-0.04	-0.01	-0.02	-0.01	-0.02	-0.02	-0.01	-0.01	-0.02
	3	-0.03	-0.01	-0.01	-0.03	-0.01	-0.02	-0.01	-0.01	-0.01
	4	-0.01	-0.03	-0.03	-0.02	-0.01	-0.01	-0.02	-0.01	0
	5	-0.02	0	-0.02	-0.01	-0.02	-0.01	-0.01	0	0
BLIP2-flan-t5-xxl	1	0	0	0.01	0	0	0.01	0	0	0
	2	0	-0.01	-0.01	0	-0.01	0	-0.02	-0.01	-0.01
	3	0	0	0	0.01	0	0	-0.01	-0.01	0
	4	0.01	-0.01	-0.02	-0.01	-0.01	0	0	0	-0.01
	5	0.01	-0.03	-0.03	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
mPLUG-Owl	1	0.06	0.03	0.02	0.02	0.02	0.03	0.03	0.02	0.01
	2	0.02	0.01	0.04	0.04	0.03	0.02	0.03	0.02	0.02
	3	0.02	0.01	0.04	0.02	0.01	0.01	0.02	0.02	-0.01
	4	0.07	0.02	0.04	0.02	0.02	0.02	0.02	0.01	0.01
	5	0.09	0.04	0.04	0.04	0.04	0.05	0.04	0.03	0.03
OpenFlamingo	1	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.02	0.02
	2	-0.01	-0.02	-0.03	-0.01	-0.02	-0.01	-0.02	-0.01	-0.01
	3	-0.03	-0.02	0	0	0	0	0	-0.01	-0.01
	4	0.01	0	0	0	0.01	0.01	-0.01	-0.01	0
	5	-0.01	-0.02	-0.01	0.01	0.01	0	-0.01	-0.01	-0.01

Table D10: Relative difference between Micro-F1 scores for the MIT Restaurant dataset when subtracting the underlying LLM from the MLLM. Score averages are for models with their corresponding prompts and amount of random N-singular few-shot examples.

Model	Prompt	Zero-shot	1-shot	2-shot	3-shot	4-shot	5-shot	6-shot	7-shot	8-shot
FROMAGe	1	0.02	-0.01	0.01	0.01	0.02	0.01	0.01	-0.01	0.01
	2	0.04	-0.01	-0.01	-0.01	0.01	-0.02	0	0	0.01
	3	0.02	0	0	0	0	-0.02	0	0.01	0.01
	4	0.06	0.01	0.03	0.04	0.04	0.02	0.03	0.01	0.01
	5	0.24	-0.01	0.02	0.03	0.03	0.01	0.04	0.02	0.03
BLIP2-opt-2.7b	1	-0.01	-0.04	-0.05	-0.07	-0.05	-0.03	-0.05	-0.02	-0.04
	2	-0.07	-0.04	-0.07	-0.1	-0.08	-0.07	-0.08	-0.09	-0.07
	3	-0.01	-0.03	-0.06	-0.07	-0.06	-0.06	-0.05	-0.07	-0.07
	4	-0.05	-0.02	-0.07	-0.09	-0.07	-0.06	-0.06	-0.05	-0.04
	5	-0.06	-0.02	-0.08	-0.12	-0.1	-0.09	-0.09	-0.07	-0.04
BLIP2-opt-6.7b	1	0	0	0.01	0.02	0.04	0.03	0.02	-0.01	-0.03
	2	0.01	0.01	-0.01	0.03	0	0.01	0	-0.01	-0.03
	3	-0.01	0.01	0	0.01	0	0.01	0	-0.02	-0.04
	4	0.05	0	0.01	0.03	0.04	0.04	0.03	0.01	-0.03
	5	0.12	-0.02	0.03	0.07	0.06	0.06	0.06	0.04	0.02
BLIP2-flan-t5-xl	1	0.02	0.02	0.02	0.01	0.01	0.02	0.01	0.01	0
	2	-0.01	0.01	0.02	0.02	0.02	0.02	0.01	0.02	0.02
	3	-0.01	0.02	0.02	0.03	0.02	0.01	0.02	0.01	0.02
	4	-0.02	0	0	-0.01	0	0	0	-0.01	0
	5	-0.03	0.02	0	0	0	0	0	0.01	0.01
BLIP2-flan-t5-xxl	1	0	0.01	0	-0.01	-0.01	-0.01	-0.01	-0.01	0
	2	0	0.01	-0.01	0	-0.01	0	0	0	0
	3	-0.04	-0.01	-0.01	-0.01	-0.01	0	-0.02	-0.01	-0.01
	4	-0.03	-0.01	-0.01	0	-0.01	-0.01	-0.01	-0.01	0
	5	-0.03	0.01	-0.01	0	-0.01	0	-0.01	0	0
mPLUG-Owl	1	0.14	0.03	0.08	0.06	0.02	0.03	0.03	0.03	0.02
	2	0.17	0.01	0.07	0.06	0.07	0.07	0.07	0.06	0.06
	3	0.19	0.02	0.06	0.04	0.04	0.04	0.04	0.03	0.04
	4	0.13	0.03	0.07	0.07	0.06	0.06	0.05	0.04	0.03
	5	0.15	0.05	0.08	0.09	0.08	0.08	0.08	0.08	0.06
OpenFlamingo	1	-0.09	0	-0.02	-0.01	-0.02	-0.04	-0.03	-0.02	-0.04
	2	-0.06	-0.01	-0.09	-0.09	-0.1	-0.13	-0.11	-0.09	-0.1
	3	-0.06	-0.03	-0.05	-0.08	-0.08	-0.13	-0.09	-0.09	-0.08
	4	-0.14	-0.01	-0.06	-0.07	-0.08	-0.12	-0.09	-0.08	-0.08
	5	-0.13	-0.02	-0.11	-0.1	-0.12	-0.16	-0.12	-0.11	-0.12

Table D11: Relative difference between Micro-F1 scores for the MIT Movie dataset when subtracting the underlying LLM from the MLLM. Score averages are for models with their corresponding prompts and amount of random N-singular few-shot examples.

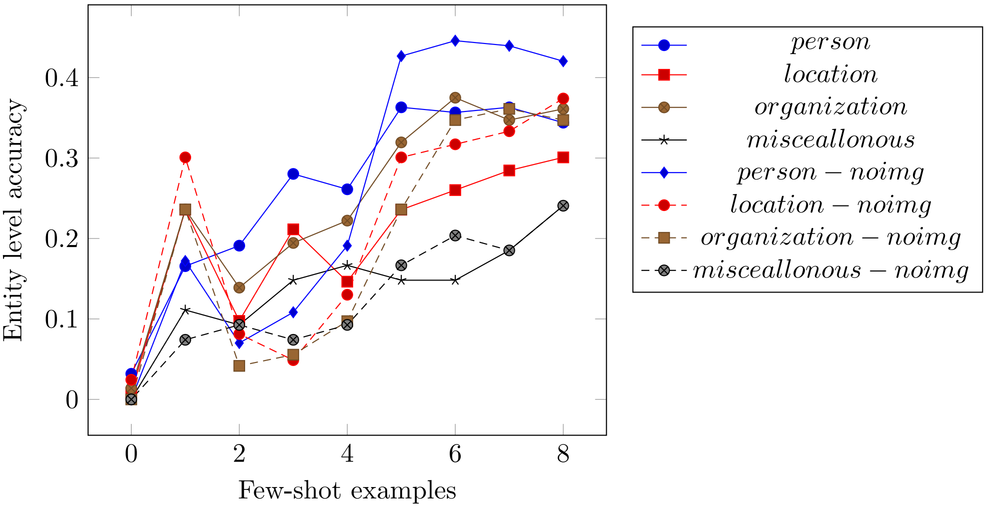

Figure 1: Accuracy per entity for Twitter2015 with BLIP2-OPT-2.7B using prompt no. 2 with and without images. With images is the original in the legend, while without is the specified by "noimg". We observe the relative changes in entity accuracy. This generalizes most of the results using OPT models.

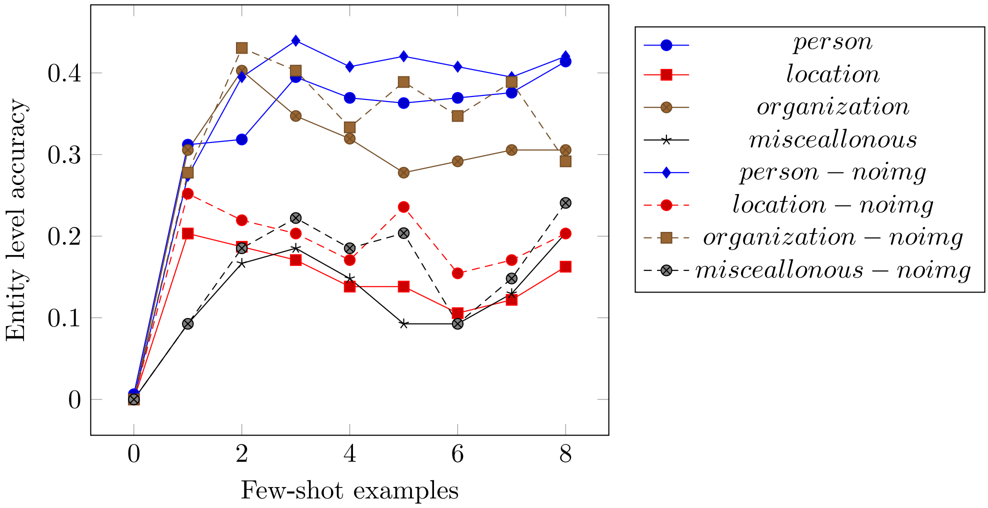

Figure 2: Accuracy per entity for Twitter2015 with BLIP2-OPT-6.7B using prompt no. 2 with and without images. With images is the original in the legend, while without is the specified by ”noimg”. We observe the relative changes in entity accuracy. This generalizes most of the results using OPT.

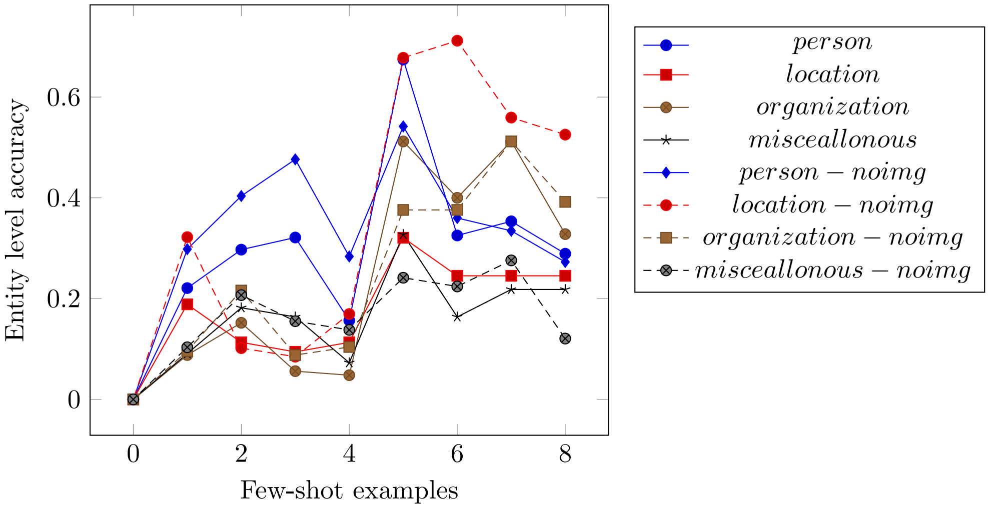

Figure 3: Accuracy per entity for Twitter2017 with BLIP2-OPT-6.7B using prompt no. 2 with and without images. With images is the original in the legend, while without is the specified by ”noimg”. We observe the relative changes in entity accuracy. This generalizes most of the results using OPT.

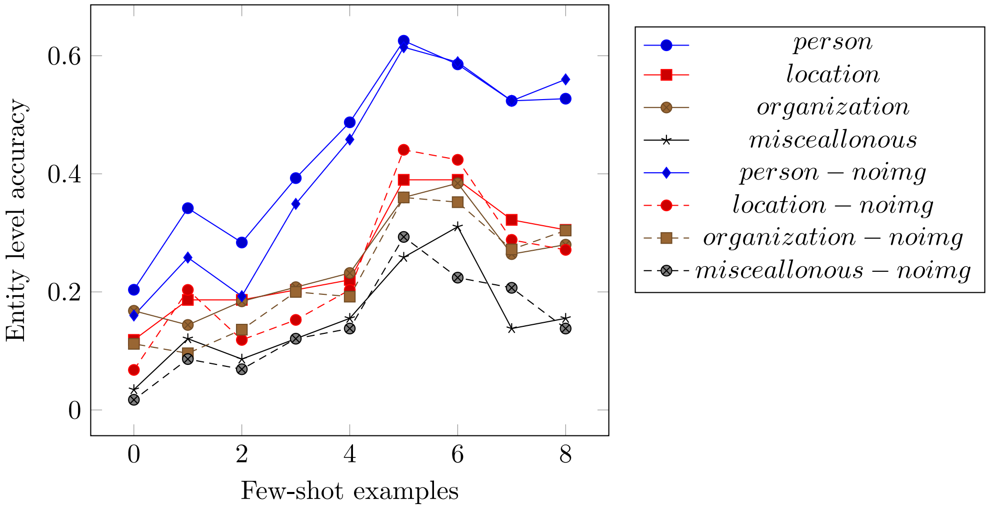

Figure 4: Accuracy per entity for Twitter2017 with OpenFlamingo using prompt no. 2 with and without images. With images is the original in the legend, while without is the specified by ”noimg”. We observe the relative changes in entity accuracy.

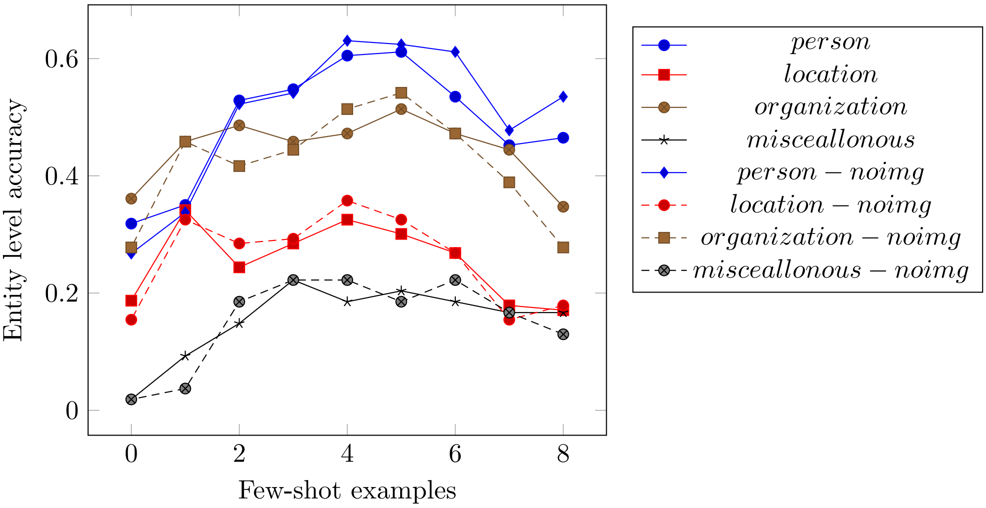

Figure 5: Accuracy per entity for Twitter2015 with OpenFlamingo using prompt no. 2 with and without images. With images is the original in the legend, while without is the specified by "noimg". We observe the relative changes in entity accuracy.

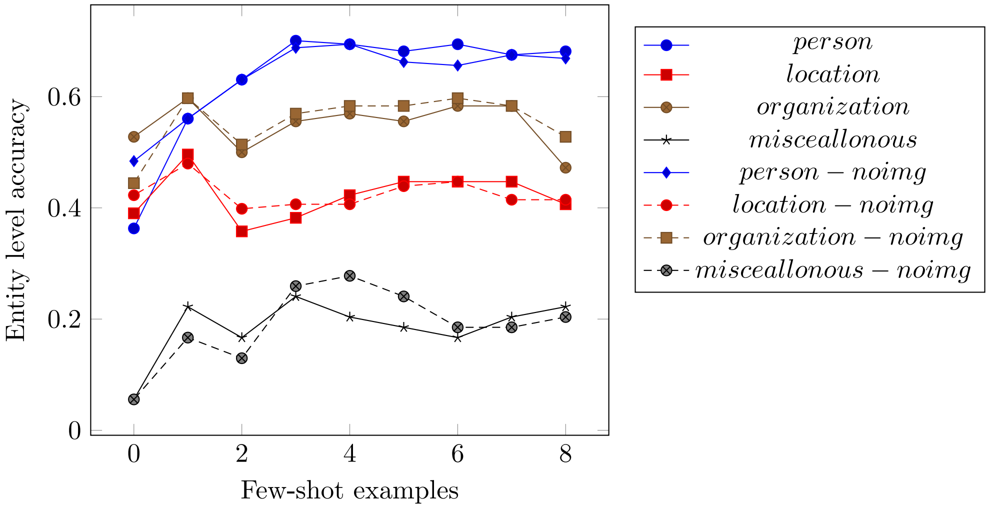

Figure 6: Accuracy per entity for Twitter2015 with mPLUG-Owl using prompt no. 2 with and without images. With images is the original in the legend, while without is the specified by "noimg". We observe the relative changes in entity accuracy.

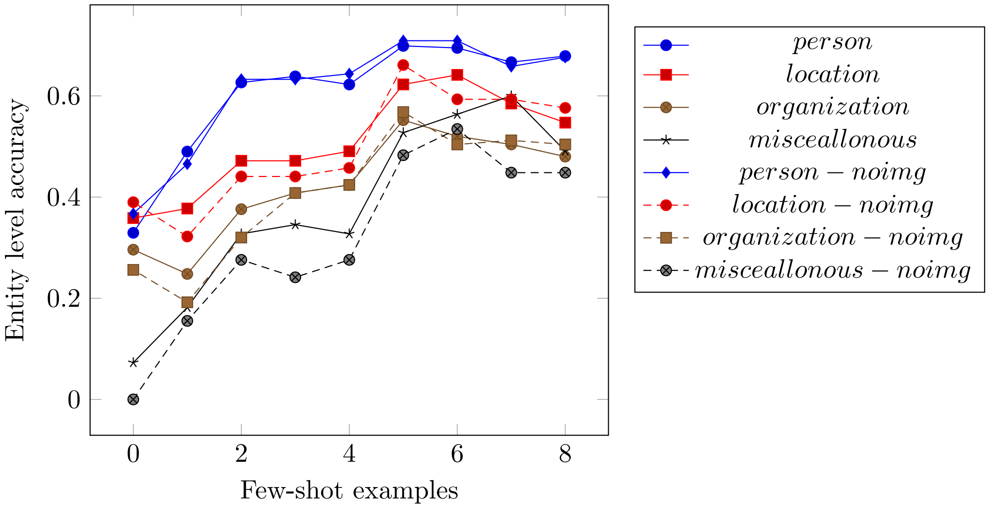

Figure 7: Accuracy per entity for Twitter2017 with mPLUG-Owl using prompt no. 2 with and without images. With images is the original in the legend, while without is the specified by "noimg". We observe the relative changes in entity accuracy.

E Deviations from the mean in unimodal results

CONLL	Prompt	Zero-shot	1-shot	2-shot	3-shot	4-shot	5-shot	6-shot	7-shot	8-shot
opt-2.7b	1	0.00	0.02	0.02	0.03	0.03	0.03	0.03	0.04	0.04
	2	0.00	0.01	0.02	0.02	0.02	0.03	0.02	0.03	0.04
	3	0.00	0.01	0.02	0.03	0.02	0.03	0.02	0.03	0.03
	4	0.00	0.03	0.03	0.02	0.03	0.04	0.03	0.04	0.04
	5	0.00	0.02	0.02	0.03	0.03	0.03	0.03	0.04	0.04
blip2-flan-t5-xxl	1	0.00	0.03	0.03	0.03	0.01	0.00	0.02	0.02	0.01
	2	0.00	0.01	0.01	0.01	0.01	0.03	0.01	0.01	0.03
	3	0.00	0.01	0.01	0.01	0.01	0.00	0.01	0.00	0.02
	4	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.01	0.01
	5	0.00	0.00	0.01	0.01	0.02	0.02	0.01	0.00	0.00
blip2-flan-t5-xl	1	0.00	0.00	0.01	0.01	0.01	0.02	0.02	0.02	0.02
	2	0.00	0.00	0.02	0.01	0.02	0.01	0.01	0.01	0.01
	3	0.00	0.00	0.01	0.02	0.02	0.02	0.03	0.02	0.03
	4	0.00	0.00	0.01	0.01	0.01	0.02	0.02	0.02	0.03
	5	0.00	0.04	0.00	0.01	0.02	0.02	0.01	0.01	0.01
mplug-owl-llama-7b-ft	1	0.00	0.04	0.03	0.03	0.02	0.03	0.03	0.03	0.03
	2	0.00	0.05	0.04	0.03	0.03	0.02	0.02	0.02	0.03
	3	0.00	0.05	0.04	0.03	0.03	0.02	0.02	0.02	0.03
	4	0.00	0.03	0.03	0.03	0.03	0.02	0.04	0.03	0.03
	5	0.00	0.03	0.04	0.03	0.03	0.04	0.05	0.04	0.04
flan-t5-xl	1	0.00	0.01	0.02	0.00	0.02	0.02	0.02	0.02	0.02
	2	0.00	0.01	0.01	0.00	0.02	0.01	0.01	0.01	0.02
	3	0.00	0.01	0.01	0.00	0.02	0.02	0.02	0.02	0.03
	4	0.00	0.00	0.00	0.01	0.01	0.02	0.03	0.02	0.03
	5	0.00	0.05	0.01	0.01	0.02	0.02	0.01	0.01	0.01
llama-7b-hf	1	0.00	0.05	0.03	0.04	0.03	0.02	0.03	0.04	0.04
	2	0.00	0.04	0.05	0.05	0.04	0.04	0.03	0.04	0.04
	3	0.00	0.05	0.05	0.03	0.03	0.03	0.03	0.04	0.04
	4	0.00	0.03	0.04	0.04	0.03	0.03	0.03	0.04	0.04
	5	0.00	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04
opt-6.7b	1	0.00	0.02	0.01	0.02	0.02	0.01	0.02	0.02	0.02
	2	0.00	0.03	0.02	0.03	0.02	0.01	0.02	0.03	0.03
	3	0.00	0.01	0.01	0.03	0.02	0.02	0.02	0.03	0.02
	4	0.00	0.03	0.01	0.02	0.01	0.01	0.02	0.04	0.02
	5	0.00	0.04	0.02	0.02	0.01	0.00	0.03	0.03	0.02
OpenFlamingo	1	0.00	0.03	0.01	0.03	0.03	0.02	0.03	0.02	0.02
	2	0.00	0.04	0.03	0.04	0.03	0.01	0.03	0.04	0.04
	3	0.00	0.02	0.01	0.05	0.02	0.02	0.03	0.05	0.03
	4	0.00	0.04	0.02	0.02	0.02	0.02	0.03	0.05	0.03
	5	0.00	0.06	0.03	0.03	0.02	0.01	0.04	0.05	0.03
flan-t5-xxl	1	0.00	0.02	0.03	0.02	0.01	0.01	0.02	0.03	0.00
	2	0.00	0.01	0.01	0.01	0.01	0.03	0.01	0.01	0.03
	3	0.00	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01
	4	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	5	0.00	0.01	0.01	0.02	0.02	0.02	0.01	0.00	0.01
blip2-opt-2.7b	1	0.00	0.03	0.04	0.05	0.03	0.06	0.04	0.04	0.05
	2	0.00	0.02	0.03	0.03	0.07	0.08	0.06	0.05	0.06
	3	0.00	0.01	0.03	0.03	0.06	0.07	0.04	0.04	0.06
	4	0.00	0.02	0.02	0.03	0.06	0.08	0.05	0.06	0.06
	5	0.00	0.02	0.05	0.03	0.07	0.07	0.05	0.05	0.05
blip2-opt-6.7b	1	0.00	0.01	0.02	0.01	0.03	0.04	0.02	0.01	0.01
	2	0.00	0.02	0.01	0.01	0.02	0.03	0.04	0.03	0.03
	3	0.00	0.01	0.01	0.00	0.02	0.03	0.04	0.04	0.02
	4	0.00	0.03	0.01	0.01	0.02	0.04	0.01	0.02	0.01
	5	0.00	0.02	0.01	0.00	0.03	0.04	0.02	0.03	0.02
fromage	1	0.00	0.02	0.03	0.03	0.04	0.03	0.01	0.02	0.02
	2	0.00	0.03	0.02	0.02	0.02	0.02	0.01	0.03	0.01
	3	0.00	0.02	0.02	0.02	0.02	0.01	0.01	0.04	0.02
	4	0.00	0.02	0.02	0.03	0.04	0.03	0.03	0.03	0.02
	5	0.00	0.02	0.02	0.03	0.04	0.04	0.01	0.02	0.02

Table E12: Table of standard deviations from the mean for the CONLL dataset experiment.

MiT Restaurant	**Prompt**	**Zero-shot**	**1-shot**	**2-shot**	**3-shot**	**4-shot**	**5-shot**	**6-shot**	**7-shot**	**8-shot**
opt-2.7b	1	0.00	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.01
	2	0.00	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.02
	3	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	4	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02
	5	0.00	0.00	0.01	0.02	0.01	0.01	0.01	0.02	0.01
blip2-flan-t5-xxl	1	0.00	0.01	0.00	0.01	0.00	0.02	0.00	0.01	0.02
	2	0.00	0.03	0.02	0.03	0.04	0.01	0.02	0.02	0.02
	3	0.00	0.02	0.01	0.03	0.02	0.02	0.02	0.02	0.01
	4	0.00	0.01	0.01	0.02	0.01	0.02	0.01	0.02	0.01
	5	0.00	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01
blip2-flan-t5-xl	1	0.00	0.01	0.00	0.02	0.02	0.01	0.02	0.01	0.01
	2	0.00	0.01	0.02	0.03	0.04	0.02	0.02	0.03	0.03
	3	0.00	0.01	0.01	0.02	0.02	0.02	0.01	0.02	0.02
	4	0.00	0.00	0.00	0.01	0.01	0.02	0.01	0.01	0.01
	5	0.00	0.01	0.00	0.01	0.02	0.01	0.01	0.01	0.01
mplug-owl-llama-7b-ft	1	0.00	0.01	0.02	0.02	0.01	0.02	0.01	0.01	0.01
	2	0.00	0.00	0.03	0.04	0.03	0.03	0.02	0.02	0.02
	3	0.01	0.01	0.02	0.03	0.02	0.01	0.01	0.01	0.00
	4	0.00	0.01	0.02	0.02	0.02	0.02	0.00	0.00	0.00
	5	0.00	0.01	0.02	0.04	0.03	0.02	0.02	0.01	0.01
flan-t5-xl	1	0.00	0.01	0.02	0.04	0.03	0.02	0.03	0.02	0.02
	2	0.00	0.01	0.02	0.04	0.03	0.02	0.02	0.04	0.03
	3	0.00	0.02	0.02	0.03	0.02	0.03	0.02	0.03	0.02
	4	0.00	0.02	0.01	0.02	0.02	0.02	0.01	0.01	0.01
	5	0.00	0.00	0.01	0.02	0.02	0.01	0.02	0.01	0.01
llama-7b-hf	1	0.00	0.01	0.02	0.02	0.01	0.01	0.01	0.01	0.01
	2	0.00	0.01	0.04	0.03	0.03	0.03	0.02	0.01	0.01
	3	0.00	0.01	0.03	0.03	0.02	0.01	0.01	0.01	0.00
	4	0.00	0.01	0.02	0.02	0.01	0.01	0.02	0.01	0.00
	5	0.00	0.01	0.03	0.03	0.03	0.02	0.01	0.01	0.01
opt-6.7b	1	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.01	0.01
	2	0.00	0.01	0.00	0.01	0.01	0.02	0.01	0.02	0.02
	3	0.00	0.02	0.01	0.00	0.01	0.01	0.01	0.01	0.02
	4	0.00	0.00	0.01	0.01	0.00	0.01	0.01	0.01	0.01
	5	0.00	0.00	0.00	0.00	0.01	0.01	0.02	0.01	0.01
OpenFlamingo	1	0.00	0.02	0.03	0.03	0.02	0.02	0.01	0.01	0.01
	2	0.00	0.03	0.03	0.03	0.02	0.02	0.01	0.00	0.00
	3	0.00	0.02	0.02	0.02	0.02	0.01	0.01	0.00	0.01
	4	0.00	0.03	0.03	0.03	0.02	0.02	0.01	0.01	0.01
	5	0.00	0.02	0.04	0.05	0.03	0.03	0.01	0.01	0.01
flan-t5-xxl	1	0.00	0.01	0.01	0.00	0.01	0.01	0.00	0.01	0.01
	2	0.00	0.03	0.02	0.03	0.03	0.02	0.03	0.02	0.02
	3	0.00	0.01	0.01	0.03	0.02	0.02	0.02	0.02	0.01
	4	0.00	0.01	0.01	0.02	0.01	0.02	0.02	0.01	0.01
	5	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
blip2-opt-2.7b	1	0.00	0.01	0.01	0.03	0.01	0.00	0.01	0.02	0.01
	2	0.00	0.01	0.02	0.01	0.02	0.01	0.02	0.01	0.02
	3	0.00	0.01	0.01	0.03	0.02	0.01	0.02	0.02	0.01
	4	0.00	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01
	5	0.00	0.01	0.02	0.03	0.02	0.02	0.01	0.01	0.02
blip2-opt-6.7b	1	0.00	0.01	0.02	0.03	0.01	0.02	0.01	0.00	0.01
	2	0.00	0.01	0.02	0.02	0.01	0.02	0.00	0.01	0.01
	3	0.00	0.02	0.02	0.02	0.01	0.02	0.00	0.01	0.00
	4	0.00	0.02	0.03	0.03	0.02	0.01	0.01	0.01	0.01
	5	0.00	0.02	0.03	0.03	0.02	0.01	0.00	0.02	0.01
fromage	1	0.00	0.02	0.02	0.03	0.01	0.01	0.00	0.01	0.00
	2	0.00	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01
	3	0.00	0.01	0.02	0.01	0.00	0.00	0.01	0.01	0.02
	4	0.00	0.03	0.03	0.03	0.02	0.01	0.01	0.01	0.02
	5	0.00	0.02	0.02	0.02	0.02	0.02	0.01	0.01	0.01

Table E13: Table of standard deviations from the mean for MiT Restaurant dataset experiment.

MiT Movie	**Prompt**	**Zero-shot**	**1-shot**	**2-shot**	**3-shot**	**4-shot**	**5-shot**	**6-shot**	**7-shot**	**8-shot**
opt-2.7b	1	0.00	0.05	0.05	0.04	0.03	0.04	0.04	0.04	0.04
	2	0.00	0.06	0.05	0.04	0.04	0.04	0.04	0.05	0.05
	3	0.00	0.05	0.04	0.04	0.04	0.04	0.04	0.05	0.05
	4	0.00	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04
	5	0.00	0.05	0.04	0.05	0.04	0.04	0.05	0.04	0.05
blip2-flan-t5-xxl	1	0.00	0.02	0.00	0.02	0.01	0.01	0.00	0.00	0.00
	2	0.00	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01
	3	0.00	0.02	0.02	0.03	0.01	0.01	0.00	0.01	0.01
	4	0.00	0.01	0.02	0.02	0.01	0.00	0.00	0.02	0.01
	5	0.00	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01
blip2-flan-t5-xl	1	0.00	0.03	0.03	0.01	0.01	0.00	0.02	0.01	0.01
	2	0.00	0.02	0.00	0.01	0.01	0.01	0.01	0.01	0.00
	3	0.00	0.02	0.01	0.01	0.01	0.01	0.02	0.01	0.01
	4	0.00	0.01	0.01	0.00	0.01	0.00	0.01	0.00	0.00
	5	0.00	0.00	0.01	0.01	0.00	0.01	0.01	0.01	0.01
mplug-owl-llama-7b-ft	1	0.00	0.07	0.03	0.02	0.01	0.02	0.05	0.04	0.05
	2	0.00	0.06	0.05	0.01	0.02	0.02	0.04	0.03	0.03
	3	0.00	0.08	0.04	0.01	0.02	0.02	0.05	0.04	0.04
	4	0.00	0.08	0.03	0.01	0.02	0.03	0.05	0.05	0.07
	5	0.00	0.09	0.05	0.01	0.02	0.03	0.05	0.05	0.06
flan-t5-xl	1	0.00	0.03	0.03	0.02	0.01	0.01	0.02	0.01	0.01
	2	0.00	0.01	0.01	0.01	0.01	0.02	0.00	0.01	0.01
	3	0.00	0.04	0.01	0.00	0.02	0.01	0.01	0.01	0.01
	4	0.00	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.00
	5	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
llama-7b-hf	1	0.00	0.05	0.04	0.03	0.03	0.04	0.04	0.04	0.05
	2	0.00	0.06	0.06	0.04	0.02	0.02	0.03	0.03	0.02
	3	0.00	0.06	0.04	0.03	0.02	0.02	0.03	0.03	0.03
	4	0.00	0.06	0.03	0.02	0.01	0.02	0.03	0.03	0.04
	5	0.00	0.08	0.04	0.02	0.02	0.03	0.04	0.04	0.04
opt-6.7b	1	0.00	0.05	0.04	0.04	0.04	0.03	0.02	0.03	0.03
	2	0.00	0.07	0.05	0.04	0.05	0.05	0.02	0.04	0.04
	3	0.00	0.06	0.05	0.04	0.05	0.05	0.03	0.03	0.04
	4	0.00	0.05	0.04	0.04	0.03	0.03	0.02	0.04	0.03
	5	0.00	0.08	0.05	0.04	0.05	0.05	0.03	0.05	0.04
OpenFlamingo	1	0.00	0.05	0.05	0.05	0.03	0.06	0.06	0.06	0.07
	2	0.00	0.05	0.05	0.05	0.03	0.02	0.02	0.03	0.03
	3	0.00	0.03	0.06	0.05	0.04	0.02	0.05	0.04	0.05
	4	0.00	0.06	0.07	0.05	0.05	0.02	0.05	0.06	0.06
	5	0.00	0.07	0.07	0.06	0.04	0.02	0.04	0.06	0.05
flan-t5-xxl	1	0.00	0.02	0.01	0.02	0.01	0.01	0.00	0.00	0.00
	2	0.00	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01
	3	0.00	0.02	0.02	0.02	0.02	0.01	0.01	0.02	0.01
	4	0.00	0.00	0.01	0.02	0.01	0.01	0.00	0.01	0.01
	5	0.00	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01
blip2-opt-2.7b	1	0.00	0.00	0.00	0.00	0.01	0.02	0.01	0.02	0.03
	2	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.02
	3	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.02	0.01
	4	0.00	0.01	0.01	0.01	0.00	0.00	0.02	0.02	0.04
	5	0.00	0.02	0.01	0.00	0.01	0.01	0.00	0.05	0.05
blip2-opt-6.7b	1	0.00	0.05	0.02	0.03	0.03	0.03	0.01	0.02	0.02
	2	0.00	0.08	0.03	0.05	0.01	0.03	0.01	0.02	0.03
	3	0.00	0.06	0.03	0.02	0.02	0.04	0.01	0.02	0.03
	4	0.00	0.05	0.03	0.03	0.03	0.05	0.03	0.04	0.03
	5	0.00	0.05	0.05	0.03	0.03	0.04	0.03	0.02	0.05
fromage	1	0.00	0.05	0.03	0.03	0.03	0.03	0.02	0.03	0.03
	2	0.00	0.06	0.04	0.01	0.03	0.03	0.03	0.03	0.02
	3	0.00	0.05	0.04	0.02	0.02	0.03	0.02	0.02	0.02
	4	0.00	0.07	0.05	0.03	0.04	0.04	0.02	0.04	0.03
	5	0.00	0.07	0.04	0.03	0.03	0.03	0.02	0.03	0.03

Table E14: Table of standard deviations from the mean for MiT Movie dataset experiment.

NTNU

Norwegian University of
Science and Technology