How Well Do Large Language Models Disambiguate Swedish Words?

Richard Johansson
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
richard.johansson@gu.se

Abstract

We evaluate a battery of recent large language models on two benchmarks for word sense disambiguation in Swedish. At present, all current models are less accurate than the best supervised disambiguators in cases where a training set is available, but most models outperform graph-based unsupervised systems. Different prompting approaches are compared, with a focus on how to express the set of possible senses in a given context. The best accuracies are achieved when human-written definitions of the senses are included in the prompts.

1 Introduction

Models of text and word meaning learned from corpus data have long played an important role in computational lexical-semantic tasks such as word sense disambiguation (WSD). Nevertheless, while previous incarnations of language representation models (e.g. count-based distributional vectors or learned static or contextual word representations) have played a crucial role in many approaches to WSD, there has been little work on applying the latest generation of language models to this task. In particular, we are aware of no previous work that investigates how well these models perform for WSD in Swedish.

In this work, we evaluate several recent large language models (LLMs) on two different evaluation sets. We consider the importance of what information is provided in the prompts: in particular, we find that the most effective prompts for most models use *definitions* of the senses, rather than just a set of related words, and we evaluate different workarounds for situations where sense definitions are unavailable.

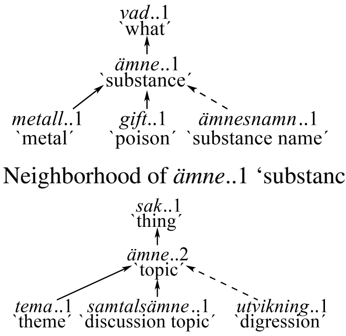
While WSD has always been something of a niche topic within the wider NLP field, it is probably useful to discuss why it is interesting at all to consider this task at the current moment. We see the importance of this study, and of WSD more generally in the present day, as twofold: 1) rather than being an intermediate step in an NLP pipeline, WSD is interesting *in itself* in a variety of use cases in lexical semantics; 2) it is also a useful for *benchmarking* the capabilities of modern language models, in particular for languages other than English.

2 The SALDO lexicon

The WSD experiments in this work are based on the SALDO lexicon (Borin et al., 2013), which defines a large sense inventory for Swedish words. This lexicon is important in Swedish NLP since it is used as a bridge between several lexical-semantic resources in Swedish (Borin et al., 2010).

As discussed by Borin and Forsberg (2009) and elsewhere, sense distinctions in SALDO are comparatively coarse-grained. Another fundamental difference to other well-known lexical-semantic resources is that SALDO does not specify typed lexical-semantic relationships (such as synonymy or hyponymy) between word senses but instead employs the concept of association (Borin et al., 2013), which can represent multiple types of lexical-semantic relationships: often, an associated sense might be a synonym or hypernym, but in other cases, it can be another relation such as a meronym.

Although each sense could theoretically have association relationships with many other senses, SALDO explicitly encodes connections between each sense and its *primary descriptor* (PD), an associated sense with a more primitive meaning. Additional relationships are (more irregularly) encoded as *secondary descriptors*. Apart from these relations, SALDO does not include any other lexical-semantic information, such as sense definitions or contextual examples. Figure 1 illustrates the neighborhoods in the SALDO graph around two senses of the noun *ämne* (‘substance’ or ‘topic’).



(a) Neighborhood of *ämne*..1 'substance'.

(b) Neighborhood of *ämne*..2 'topic'.

Figure 1: Fragments of SALDO neighborhoods for two of the senses of *ämne*. Primary descriptor edges are drawn as solid arrows and secondary descriptor edges as dashed arrows.

3 Method

3.1 Datasets and Experimental Setup

The experiments are carried out on two different sense-annotated datasets.

SENSEVAL-2 The largest sense-annotated resource for Swedish was developed in the SemTag project (Järborg, 1999); this covers most of the SUC corpus (Ejerhed et al., 1992). The Swedish lexical sample of the *SENSEVAL-2* shared task (Kokkinakis et al., 2001) is a subset of the SUC dataset and included annotated instances for 40 ambigous lemmas. This dataset does not use the SALDO sense inventory, but the senses for these lemmas were manually mapped to SALDO by Nieto Piña and Johansson (2016). Since SALDO uses a coarser division into senses than SemTag, three of the lemmas were not ambiguous after the conversion and were removed from the dataset.

Eukalyptus The only running-text corpus annotated with SALDO senses is *Eukalyptus* (Johansson et al., 2016), which has texts from eight domains.

Preprocessing The instances were preprocessed using the *Sparv* pipeline (Borin et al., 2016). For each word, the pipeline proposes a set of possible SALDO senses, based on an automatic morphological analysis and lemmatization.

For *Eukalyptus*, unambiguous words were excluded from the experiment. This means that the *practical* accuracy is higher than what we report in the next section, since the majority of the words are unambiguous. We also exclude cases where the annotated sense is a non-compositional reading of a multi-word expression (e.g. *på örat* intended as 'drunk', not as 'on the ear') or a compositional reading of a compound. After preprocessing the two datasets, SENSEVAL-2 consists of a test set of 1,366 instances and a training set of 7,790 instances, and the *Eukalyptus* set of 12,434 instances.

Experimental setup During evaluation, a disambiguator is given an instance that includes the target word and a five-sentence context centered on the target word. (Using larger contexts did not impact results meaningfully except in terms of cost and processing time.) The disambiguator then chooses one of the alternatives proposed by the Sparv lemmatization pipeline. For SENSEVAL-2, the reported accuracy is a macro-average over the 37 lemmas (20 nouns, 11 verbs, 6 adjectives). For *Eukalyptus*, we report a micro-averaged accuracy computed over all instances in the corpus.

3.2 Baselines

Before evaluating the LLM-based disambiguators, we ran a number of trivial and nontrivial baselines on the same datasets. Tables 1 and 2 show the results on the SENSEVAL-2 and *Eukalyptus* benchmarks, respectively. The *Random* accuracy corresponds to the mean accuracy achieved when selecting a sense uniformly randomly from the set of alternatives. The *First sense* baseline selects the the sense with the lowest numerical identifier among the alternatives; while SALDO senses are not ordered by frequency, the most common senses are often listed earlier, so this can be seen as a proxy of a most-frequent-sense baseline. We also include *Upper bound* here: the accuracy we'd get if always selecting the gold-standard sense if it is among the alternatives. Since there are occasional mistakes in lemmatization and the list does not always include the true sense, this is not exactly 1.0.

Among nontrivial baselines, we evaluated various *graph-based unsupervised* and *supervised* methods. Graph-based unsupervised methods use the SALDO graph in different ways but are not trained on a sense-annotated dataset. In this category, we evaluate three different systems: *Personalized PageRank* uses a graph algorithm to disambiguate (Agirre and Soroa, 2009); *Sense vectors* is the method by Johansson and Nieto Piña (2015), which uses the SALDO graph to create vector representations of senses. the *BERT substitutes* method (Johansson, 2022) uses a BERT model to propose words that could be substituted for the target word, and then compares the substitute set to the graph

System	Accuracy
Random	0.349
First sense	0.495
Upper bound	0.992
Personalized PageRank	0.497
Sense vectors	0.498
BERT substitutes	0.668
BERT + LR	0.931
BoW + SVM	0.808

Table 1: Macro-averaged accuracies on the SENSEVAL-2 test set for three categories of baselines: trivial, graph-based unsupervised, and supervised models.

System	Accuracy
Random	0.402
First sense	0.658
Upper bound	0.992
BERT-based substitutes	0.702

Table 2: Micro-averaged accuracies on the *Eukalyptus* test set for two categories of baselines: trivial and graph-based unsupervised models.

neighborhoods for the different senses.

Among *supervised* methods, we evaluated two approaches: a linear SVM using a bag-of-words representation, and a logistic regression on a BERT output at the target token. Both were implemented as "word experts" that use one classifier per base form (Berleant, 1995). Supervised models were only evaluated for SENSEVAL-2, which comes with a train/test split; *Eukalyptus* includes many low-frequency lemmas and a supervised word expert approach would be less meaningful.

3.3 Included Models

In our experiments, we considered the following models. They were accessed through their respective APIs without any fine-tuning.

- *Claude 3.5 Sonnet* (Anthropic, 2024);
- *Command R+* (170B) (Cohere, 2024);
- *Gemini 1.5 Pro* (Google, 2024);
- *Llama 3 8B* and *70B*, and *Llama 3.1 405B* (Llama Team, 2024), via the Replicate API;[1]
- *GPT-3.5 Turbo* (Ouyang et al., 2022), *GPT-4 Turbo* (OpenAI, 2023), *GPT-4o*, and *GPT-4o-mini*.

Appendix B.4 exemplifies usage costs for some of these models.

3.4 Prompt Design

The prompts consisted of the following parts:

1. a preamble describing the WSD task: that the goal is to pick one out of a given list of senses;
2. an example demonstration;
3. a list of senses to choose from;
4. the context to disambiguate.

The investigations in the next section focus on the third point: how the senses are specified.

4 Experiments

4.1 Neighborhoods or Definitions?

In our first investigation, we investigated different ways of presenting the list of SALDO senses to the model. As mentioned in §2, the only information about senses given directly in SALDO is the set of neighbors in the sense graph, as in Figure 1.

The first prompting approach, the *neighborhood* prompting technique, specifies the sense inventory for a given lemma simply by enumerating the lemmas of up to four direct neighbors in the sense graph. For a sense s, we first list the primary descriptor of s, followed by senses for which s is the primary descriptor. We also include secondary descriptors if the number of neighbors is less than 4. For instance, the neighborhood of the first sense of *ämne* 'substance' shown in Figure 1a would be encoded as *vad, metall, gift, ämnesnamn*. Appendix B.1 shows an example in detail.

We compare the neighborhood prompting technique to a second approach where we give textual definitions of the senses. Definitions are not available in SALDO, but for the SENSEVAL-2 dataset we can rely on the mapping between SALDO and SemTag senses and use their definitions. Appendix B.2 shows how these prompts are written.

Table 3 shows the results on the SENSEVAL-2 test set for all models with the two prompting techniques. To give an impression of the variation in accuracy across the 37 lemmas, detailed results for the best LLM, a trivial baseline, and the best supervised model are presented in Appendix A. The takeaways from this evaluation are: 1) *all* 10 LLMs perform better with definition prompts than with neighborhoods, although there is some varition in the performance gap between the two approaches; 2) unsurprisingly, newer models outperform older models and larger models perform smaller models; 3) most LLMs outperform the graph-based unsupervised baselines; 4) no LLM reaches the level of the best supervised baseline.

[1] https://replicate.com/home

Model	N	D
claude-3-5-sonnet	0.778	0.855
gpt-4o	0.792	0.849
llama-3.1-405b-instruct	0.728	0.818
gpt-4-turbo	0.745	0.805
llama-3-70b-instruct	0.699	0.767
gemini-1.5-pro	0.730	0.737
gpt-4o-mini	0.676	0.727
gpt-3.5-turbo	0.466	0.551
command-r-plus	0.431	0.527
llama-3-8b-instruct	0.363	0.522

Table 3: SENSEVAL-2 accuracies for all models. We compare prompts based on SALDO neighborhoods (N) and definitions (D).

Model	N	AD	CoT
gpt-4o	0.792	0.792	0.788
llama-3.1-405b-instruct	0.728	0.788	0.758
llama-3-70b-instruct	0.699	0.767	0.737
gpt-4o-mini	0.676	0.695	0.739

Table 4: SENSEVAL-2 accuracies with neighborhoods (N), automatically generated definitions (AD), and chain-of-thought prompts (CoT).

4.2 Model-written Sense Definitions

The results in §4.1 showed that all LLMs perform better with explicit definitions of senses rather than implicit specifications via SALDO neighborhoods. However, since the SALDO–SemTag mapping is only definied for the 37 lemmas in the SENSEVAL-2 benchmark, we are unable to use definition-based prompts in the wild. But could we work with definitions written *automatically* by a model instead of lexicographer-written definitions?

We implemented this idea in two different ways. First, we let GPT-4o write automatically generated sense definitions (AD) based on SALDO neighborhoods and used them as in the previous definition-based prompts. (Appendix B.3 shows the prompt we used to write the definitions.) We did not evaluate definitions written by any other model. Secondly, we used a *chain-of-thought* (CoT) approach (Wei et al., 2022) where the model is given the SALDO neighborhoods of the senses for a given instance, and is asked to write definitions before selecting the sense identifier. For reasons of cost and computational efficiency, we only considered a subset of models in this investigation.

Table 4 shows the SENSEVAL-2 results. The best-perfoming model (GPT-4o) is not improved by any of these techniques: the scores are more or less unchanged from the baseline (neighborhood prompts). For the weaker models, both variants of definition-writing approaches seem to improve over the baseline. There is no consistent difference over the set of models between AD and CoT.

Model	N	AD
gpt-4o	0.852	0.818
llama-3.1-405b-instruct	0.826	0.823
llama-3-70b-instruct	0.781	0.799
gpt-4o-mini	0.756	0.756

Table 5: *Eukalyptus* accuracies with neighborhoods (N) and automatically generated definitions (AD).

In addition, we compared neighborhoods and GPT-4o-written definitions on the larger *Eukalyptus* dataset, and Table 5 shows the accuracies. CoT is not evaluated here because of its higher cost: with AD, each sense only has to be defined once. Compared to SENSEVAL-2, *Eukalyptus* has a different distribution of lemmas and senses, and the results are different from those in Table 4. While the relative ranking of models is the same, it does not seem to be useful to include automatically written definitions in the prompts (except for Llama-70B).

5 Conclusions

To answer the question posed by the title of this paper, current LLM word sense disambiguators without any fine-tuning perform at a level between unsupervised graph-based and supervised disambiguators for the Swedish datasets we considered. All LLMs perform better in this task when textual *definitions* are used to specify the senses, as opposed to just listing a set of related words; however, even with the neighborhood-based approach, disambiguation accuracy is still substantially better than with unsupervised approaches for most LLMs. Definitions written automatically seem less useful than human-written definitions and do not consistently outperform neighborhood-based prompts.

Based on these results, we can conclude that most recent LLMs have some capability of carrying out a task that requires a fairly fine-grained semantic representation of passages in Swedish. It would be useful to consider the amount of Swedish included in pre-training and instruction-tuning for these models, and how these quantities relate to the performance figures reported here, but as of now this information is unavailable.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 33–41, Athens, Greece.

Anthropic. 2024. Claude 3.5 Sonnet model card addendum.

Daniel Berleant. 1995. Engineering "word experts" for word disambiguation. *Natural Language Engineering*, 1:339–362.

Lars Borin, Dana Dannélls, Markus Forsberg, Maria Toporowska Gronostaj, and Dimitrios Kokkinakis. 2010. The past meets the present in the Swedish FrameNet++. In *Proceedings of EURALEX*.

Lars Borin and Markus Forsberg. 2009. All in the family: A comparison of SALDO and WordNet. In *Proceedings of the Nodalida 2009 Workshop on WordNets and other Lexical Semantic Resources - between Lexical Semantics, Lexicography, Terminology and Formal Ontologies. NEALT Proceedings Series*, volume 7.

Lars Borin, Markus Forsberg, Martin Hammarstedt, Dan Rosén, Roland Schäfer, and Anne Schumacher. 2016. Sparv: Språkbanken's corpus annotation pipeline infrastructure. In *Swedish Language Technology Conference*, Umeå, Sweden.

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2013. SALDO: a touch of yin to WordNet's yang. *Language Resources and Evaluation*, 47(4):1191–1211.

Cohere. 2024. Command R+.

Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. The linguistic annotation system of the Stockholm-Umeå corpus project – description and guidelines. Technical report, Department of Linguistics, Umeå University.

Gemini Team, Google. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Richard Johansson. 2022. Coveting your neighbor's wife: Using lexical neighborhoods in substitution-based word sense disambiguation. In Elena Volodina, Dana Dannélls, Aleksandrs Berdicevskis, Markus Forsberg, and Shafqat Virk, editors, *LIVE and LEARN – Festschrift in honor of Lars Borin*, pages 61–66. University of Gothenburg, Gothenburg, Sweden.

Richard Johansson, Yvonne Adesam, Gerlof Bouma, and Karin Hedberg. 2016. A multi-domain corpus of Swedish word sense annotation. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 3019–3022, Portorož, Slovenia.

Richard Johansson and Luis Nieto Piña. 2015. Combining relational and distributional knowledge for word sense disambiguation. In *Proceedings of the 20th Nordic Conference of Computational Linguistics*, pages 69–78, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Jerker Järborg. 1999. Lexikon i konfrontation. Technical report, University of Gothenburg. Research Reports from the Department of Swedish, Språkdata, GU-ISS-99-6.

Dimitrios Kokkinakis, Jerker Järborg, and Yvonne Cederholm. 2001. SENSEVAL-2: The Swedish framework. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 45–48, Toulouse, France.

AI @ Meta Llama Team. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Luis Nieto Piña and Richard Johansson. 2016. Benchmarking word sense disambiguation systems for Swedish. In *Swedish Language Technology Conference*, Umeå, Sweden.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeff Wu, and Xu Jiang et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

A Example of Detailed SENSEVAL-2 Results

Table 6 shows the evaluation scores for each lemma in the SENSEVAL-2 test set of the best-performing LLM (Claude 3.5 Sonnet) with definition-based prompts, the first-sense baseline, and the best-performing supervised system (BERT with logistic regression).

Lemma	N	Claude	Baseline	BERT+LR
barn	115	0.8348	0.5826	0.8783
betydelse	52	1.0000	0.1538	0.9808
bred	18	1.0000	0.3889	1.0000
flytta	32	0.8438	0.3125	0.9375
fylla	11	1.0000	1.0000	1.0000
färg	19	0.9474	0.6316	0.8947
förklara	30	0.9000	0.5667	0.9000
gälla	148	0.7635	0.6216	0.9459
handla	44	1.0000	0.1136	1.0000
höra	92	0.8696	0.3370	0.8696
klar	54	0.9259	0.1296	0.9630
konst	13	0.6923	0.4615	0.7692
kraft	20	0.9000	0.8500	1.0000
kyrka	27	0.9259	0.7037	0.9630
känsla	25	0.6400	0.6800	0.9600
ledning	16	0.4375	0.5625	1.0000
makt	21	1.0000	1.0000	1.0000
massa	16	0.9375	0.5625	0.9375
mening	28	1.0000	0.5357	0.9643
måla	16	0.8125	0.2500	0.8750
natur	16	1.0000	0.1250	0.9375
naturlig	24	0.8750	0.3750	0.8750
program	24	0.5417	0.5000	1.0000
rad	25	0.8000	0.2400	0.9200
rum	39	0.9487	0.8974	1.0000
scen	17	0.8235	0.7059	0.9412
skjuta	12	0.7500	0.3333	0.8333
spela	38	0.8947	0.2895	0.9211
stark	62	0.5968	0.4194	0.8387
tillfälle	20	0.9500	0.6000	0.9500
uppgift	30	1.0000	0.1667	0.9667
vatten	50	0.8800	0.8000	0.9800
vänta	43	0.7907	0.6512	0.9070
ämne	34	0.9118	0.1765	0.8529
öka	77	0.7792	0.6364	0.9610
öppen	33	0.8485	0.2424	0.7576
öppna	25	0.8000	0.7200	0.9600

Table 6: Detailed evaluation scores on the SENSEVAL-2 test set.

B Prompts

B.1 Neighborhood-based prompt

```
SYSTEM PROMPT:
```

You are a tool that carries out word sense disambiguation in Swedish. In the following, you will be given sentences where your task is to disambiguate the word surrounded by asterisks (*). You should select a sense identifier from a given list of senses. Each sense is specified by a set of related words. Based on the related words, first write definitions of the senses, then explain how the example relates to one of the senses, and finally output the relevant sense of the word in the context.

Answer with one of the sense identifiers, or 0 if none of them are applicable. The last line of the output must correspond to the sense identifier and include no other text.

Example input:
Entry: rock
Senses:
1: related to "kappa", "bilrock", “bonjour”; 2: related to "musik", "hårdrock", "indierock".
Sentence: Bandet spelade * rock * .
Output:
2

USER PROMPT:
Entry: öppna
Senses: 1: related to "öppen", "bryta", "bryta upp", "dekantera";
2: related to "starta", "öppnande", "verksamhet"
Sentence: " Ja , du får nog göra det " , sa en av dem . Söderberg såg oförstående på honom . " Du får nog ta och * öppna * " , sa mannen igen . Söderberg ville krympa ihop och bli liten . Så liten att han försvann .

B.2 Definition-based prompt

SYSTEM PROMPT:
You are a tool that carries out word sense disambiguation in Swedish. In the following, you will be given sentences where your task is to disambiguate the word surrounded by asterisks (*). You should select a sense identifier from a given list of senses. Based on the definitions of the senses, first explain how the example relates to one of the senses, and finally output the relevant sense of the word in the context.
Answer with one of the sense identifiers, or 0 if none of them are applicable. The last line of the output must correspond to the sense identifier and include no other text.

Example input:
Entry: rock
Sense definitions:
1: ytterplagg för överkroppen, som räcker ungefär till knäet och har ärmar
2: typ av melodisk enkel musik i kraftfullt markerad 4/4-takt
Sentence: Bandet spelade * rock * .
Output:
2

USER PROMPT:
Entry: öppna
Sense definitions:
1: bringa till öppet (eller öppnare) läge; med avs. på spärrande anordning, ibl. underförstådd; äv. med tanke på utrymmet innanför, personen etc. som skall släppas in m.m.; ibl. symboliskt i liknelser
2: göra tillgänglig för användande eller utnyttjande; av större grupp personer; med avs. på

Model	Cost
gpt-4-turbo	0.0053
llama-3.1-405b-instruct (via Replicate)	0.0036
gpt-4o	0.0027
claude-3.5-sonnet	0.0026
llama-3-70b-instruct (via Replicate)	0.00037
command-r-plus	0.0016

Table 7: Costs in US dollars per API call for a subset of the models, with human-written definitions.

```
inrättning e. d., ibl. underförstådd; äv. med avs. på tidpunkt; inrätta (och påbörja) verksamhet
med; ngt slags företag e. d.; börja utföra; viss angiven el. underförstådd verksamhet
Sentence: " Ja , du får nog göra det " , sa en av dem . Söderberg såg oförstående på honom . " Du
får nog ta och * öppna * " , sa mannen igen . Söderberg ville krympa ihop och bli liten . Så liten
att han försvann .
```

B.3 Prompt for Writing Definitions

```
SYSTEM PROMPT:
You are lexicographer who writes dictionaries describing words in Swedish.

In the following, you will be given a list of senses of a given input word. For each sense, a set
of related words is given. Your task is to write a short definition in Swedish and give an example
sentence of each of the given senses.

The output should be a JSON object where each key is a sense identifier and each value a list
containing the definition and the example sentence.

Example input:
Entry: "rock"
Senses: 1: related to "kappa", "bilrock", “bonjour”; 2: related to "musik", "hårdrock",
"indierock".

Example output:
{ "1": ["Ytterplagg för överkroppen, som räcker ungefär till knäet och har ärmar.", "Han hade
på sig en lång rock."], "2": ["Typ av melodisk enkel musik i kraftfullt markerad 4/4-takt.",
"Bandet spelade rock."] }

USER PROMPT:
Entry: öppna
Senses: 1: related to "öppen", "bryta", "bryta upp", "dekantera";
2: related to "starta", "öppnande", "verksamhet"
```

B.4 API costs

Table 7 shows the usage cost for some of the LLMs evaluated in this paper. The table shows the cost in US dollars of one API call when the prompt includes human-written definitions. Neighborhood-based prompts are somewhat cheaper, and chain-of-thought prompts more expensive.