

NLP - A2

Bente van Katwijk, Tom de Haas, and Thomas Grimsgaard

April 2024

Group Contract

Members of the Group

- Bente van Katwijk / Contact Information
- Tom de Haas / Contact Information: t.de.haas2@student.vu.nl
- Thomas Grimsgaard / Contact Information

Strengths and weaknesses of each group member

Strengths

- Bente: planning/scheduling and decision making
- Thomas: Conflict resolution, communication
- Tom: pragmatism and stress resistance

Weaknesses

- Bente: communication
- Thomas: Time management
- Tom: distraction and punctuality

Overall learning goal

Our overall learning goal is to have fun and learn a lot during the coding part, as this is something we all would like to have more experience in, and have a pleasant collaboration.

Expected Behavior

Punctuality and Timeliness

1. All group members will be punctual. Meetings will begin five minutes after the appointed start time. Everyone should be present and ready at that time.
2. Attendance at all meetings is expected unless due to unavoidable circumstances like illness.
3. All members will stay for the duration of the meeting unless all tasks are completed or the meeting is adjourned unanimously.

Procedures

1. Members should come to meetings prepared, having read the assigned material and ready to discuss tasks and decisions.
2. Agreed tasks must be completed by the set deadlines. Delays must be communicated and help sought promptly.
3. At the end of each session, time is set aside to evaluate the completion of tasks and group mechanisms.
4. Each group member has the right to indicate if any of these rules are being violated.
5. Information is to be shared via short messages on platforms like WhatsApp or Teams.
6. All documents are saved and shared on Github and Overleaf in the folder (so data is always backed up). Folders are numbered and properly described.

Behavior

1. The group seeks consensus actively, with all members participating in discussions.
2. Each member will take turns listening and talking, and active listening will be a strategy for all group discussions.
3. Sexist and racist comments are unacceptable.
4. No aggressive or dominant behavior will be tolerated.

Roles

1. Roles such as leader, secretary, and timekeeper are assigned at or before each meeting.
2. The leader sets sub-goals, leads discussions, and presents group outcomes.
3. The secretary takes meeting notes and prepares presentations.
4. The timekeeper monitors discussion times and informs the group about the remaining time.
5. The devil's advocate keeps an open mind to all possibilities and viewpoints.

Roles during meetings

1. Meeting 1 (22/04/2024):
 - Leader: Tom
 - Secretary: Bente
 - Timekeeper: Thomas
 - Devil's advocate: Tom
2. Meeting 2 (25/04/2024):
 - Leader: Bente
 - Secretary: Tom
 - Timekeeper: Bente
 - Devil's advocate: Thomas

Evaluate

At the beginning of every meeting there is some time allotted for evaluations if needed, to ensure adherence to the contract and to discuss necessary adjustments.




Method for Resolving an Impasse

1. Identify and try to reach consensus on points of disagreement. Proceed to step 2 if no consensus is reached.
2. The leader assesses the conflict's relevance and may defer it if its importance is limited.
3. The leader will decide how much time there is for discussion or arbitration before proceeding to a vote.
4. The leader will call a vote. If the vote results in a deadlock, the leader will make a final decision.

Table 1: Timeline and task allocation

Deadline	Exercises	Responsible group member
Mon. 22/04/2024	Ex. 0	Thomas and Tom
	Ex. 1	Bente
	Ex. 2	Thomas
	Ex. 3	Bente
Tue. 23/04/2024	Ex. 4	Tom and Bente
	Ex. 5	Thomas
Wed. 24/04/2024	Ex. 6	Bente and Thomas
	Ex. 7	Tom
	Bonus	Thomas
Thu. 25/04/2024	Final touch and discussion of results	Everyone

Table 2: Signatures by all team members on 22/04/2024

Bente van Katwijk	Tom de Haas	Thomas Grimsgaard
		

1 Answers to written QAs

Exercise 0: Empirically verify Zipf's law on the Brown corpus

Zipf's law can be verified when comparing the specific genre parts of the corpus versus the total size of the corpus. The number of word types increases proportionally with total number of words. See table 3 at the bottom of the document.

Exercise 1: Creating the `word_to_index` dictionary

Exercise 2: Building an MLE unigram model

Written QA

Eyeballing the proportion of word types occurring only once, we estimate between 50% and 75%, based on visually inspecting the printed counts-vector, as it's mostly populated by ones. The proportion of types would most likely not change with a larger corpus due to Zipf's law, which states that the most frequent word in a corpus will occur twice as often as the second most frequent, three times as often as the third, and so on. This means the least frequent words are very likely to only occur only once. In a bigger corpus, more common words would appear more frequently, but there would also new words added due the increased vocabulary. Consequently, the proportion of words occurring only once will likely stay very similar. For example, the word "the" occurs most often as it is necessary to construct grammatically correct utterances. Other words, like "hubris" or "besmirched", are not grammatically essential and are less common in every day language and therefore will occur less: only once in the Brown corpus.

Exercise 3: Building an MLE bigram model

Exercise 4: Add- α smoothing the bigram model

Written QA

Smoothing is a technique that makes sure that no probability of a word occurring after some other word is set to 0. It is especially useful to make the language model generalize better to unseen data. In this text, for example, the word 'all' appears only once and the word 'the' is followed by it, leading to $P(the | all) = 1.0$. For this text this is indeed true, but in other texts it does not make sense that the word 'the' always follows the word 'all'. By adding a small constant to each count, no probability is 0. Considering $P(the | all)$, this means that there is also a non-zero probability assigned to other words following the word 'all', even though that was not the case in the text it was trained on. Since all of the counts are a little bit higher, this means that there is a very small probability $P(word | previous_word)$ assigned to every possible word combination in the text. This leads to a decrease in other probabilities of words that actually do occur together in the text, as all the probabilities for some *previous_word* should sum to 1.

Written QA

Probabilities conditioned on words that do not occur often in a text (such as 'all', this word occurs only once in the text) fall much more than those of words that occur often (such as 'the'). When no smoothing is used, as explained in the previous answer, the probability of $P(word | previous_word)$ will be very high

for the words that occur after some *previous_word* if that *previous_word* does not occur often in a text. The more often a *previous_word* occurs in the text, the more different words there are that follow this *previous_word*, so the lower the $P(\text{word} \mid \text{previous_word})$ for these words (as they all sum to 1, and there are more different words). With smoothing, all combinations get a small probability assigned. For words that occur more often in the text, the probability already was a lot lower for the combinations of words that occur in the text, so it will drop less dramatically. This is a good thing, because these probabilities will generalize better to unseen texts, as it is unlikely that in other texts, the word 'calls' will follow 'anonymous' 33% of the time.

Exercise 6: Calculating sentence probabilities

Written QA

The perplexity of the unigram model is the highest, followed by the perplexity of the smoothed bigram model and then the normal bigram model. The lower the perplexity, the better the model performs; the better it generalizes to the test set. The perplexity of language models can only be compared if the same vocabulary was used to train the models, which was the case, as the models were trained on the same corpus. It is no surprise that the unigram yields the highest perplexity, as a bigram model knows better which words logically follow each other, so they assign those bigrams a higher probability, leading to a lower perplexity. In conclusion, we can say that the unsmoothed bigram model has the highest performance, followed by the smoothed bigram model. The unigram model had the lowest performance.

Written QA

Smoothing hurt the performance of the model, which we conclude as it has a higher perplexity than the unsmoothed bigram model. This can be explained by the fact that the sentences in the *toy corpus* are exactly the same as the first two sentences from the *Brown corpus*, on which the models were trained. This is because smoothing makes sure that the model generalizes better to unseen text, whereas an unsmoothed model is 'overfitting' the training data: the probabilities from the unsmoothed bigram model reflect exactly the probabilities of those bigrams occurring in the text. Since the models were tested on the exact same sentences that occurred in the training data, it makes sense that the model performs well on these testing sentences.

Exercise 7

Written QA

With regards to generation the three models; unigram, bigram and smoothed bigram did not perform equally. The smoothed model makes sentences that feel more like they make sense compared to the bigram and unigram model. In

term of creativity, there could be a fine line that distinguishes creative sentence forming and just plain incongruent combination of word. The unigram and bigram falling more towards the latter category, whereas the smoothed bigram generates more congruent sentences.

The unigram is not congruent with semantical rules. However, the bigram and smoothed model is. Word combinations can be found that make are quasi-meaningful. Whereas in the unigram model, word are generated at complete randomness. Even though for the smoothed model, the sentences are correct grammatically, it still does not generate any sensible meaning except on accident in the same way a broken clock is right twice a day. The smoothed model did not terminal early, thereas the bigram and unigram did. The unigram actually terminating once without even generating any words. Because meaning is lacking in the generated sentences, no point with regards to ambiguity can be made. These models do not account for any lexical semantics yet.

Bonus

Written QA

In unigram models there is an assumption of independence, meaning that the words are assumed to be independent of each other and therefore that the meaning can be gained from analysing the words alone. However, this is not necessarily a reasonable assumption and deserves investigation. Calculating the independence of the words in the corpus is a way to investigate this, which can be done by calculating the point wise mutual information (PMI) between word pairs. A high positive value means that the words occur together often and therefore can be said to not be independent. A very negative value means that the words often appear separately. When the words are independent, it can be said that the probability of one word is not affected by the previous word, and this is will have a PMI value of 0.

Table 3: Origin of Brown corpus findings

Genre	Facts	Findings
Humor	<ol style="list-style-type: none"> 1. Number of types 2. Total number of words 3. Average word length 4. Number of Tokens 5. Average number of words per sentence 6. Ten most frequent POS tags (POS, count) 	<ol style="list-style-type: none"> 1. 4,755 2. 21,695 3. 4,04 4. 21,982 5. 20.6 6. [(‘NN’, 3598), (‘IN’, 2289), (‘DT’, 1975), (‘JJ’, 1552), (‘,’, 1331), (‘VBD’, 1237), (‘.’, 1117), (‘RB’, 1104), (‘PRP’, 1048), (‘NNS’, 864)]
Science Fiction	<ol style="list-style-type: none"> 1. Number of types 2. Total number of words 3. Average word length 4. Number of Tokens 5. Average number of words per sentence 6. Ten most frequent POS tags (POS, count) 	<ol style="list-style-type: none"> 1. 3,032 2. 14,470 3. 3.99 4. 14,631 5. 15.26 6. [(‘NN’, 2158), (‘IN’, 1387), (‘DT’, 1280), (‘JJ’, 1054), (‘.’, 991), (‘VBD’, 938), (‘PRP’, 810), (‘,’, 791), (‘RB’, 772), (‘NNS’, 596)]
No specific genre	<ol style="list-style-type: none"> 1. Number of types 2. Total number of words 3. Average word length 4. Number of Tokens 5. Average number of words per sentence 6. Ten most frequent⁸ POS tags (POS, count) 	<ol style="list-style-type: none"> 1. 49,815 2. 1,161,192 3. 4.28 4. 1,173,780 5. 20.25 6. [(‘NN’, 208427), (‘IN’, 136402), (‘DT’, 116698), (‘JJ’, 94264), (‘NNS’, 61272), (‘.’, 58336), (‘,’, 56523), (‘RB’, 50152), (‘VBD’, 46243), (‘PRP’, 40580)]