**Question 5 – Character Level Embedding**

**Named Entity Recognition**

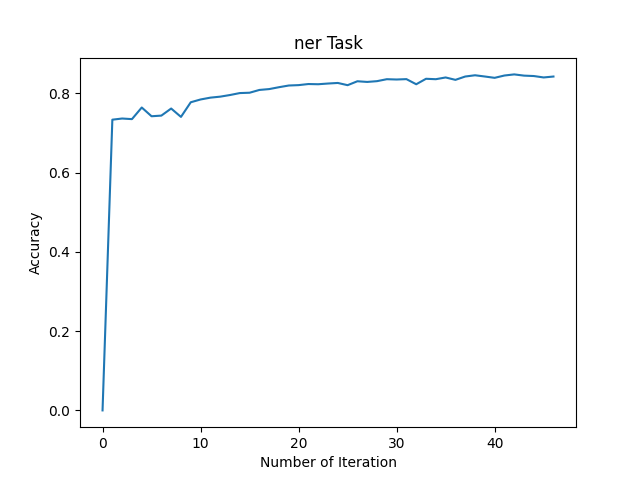
Note: max word length was 10 because 99 percentiles of the data were below that

**Experiment 1:**

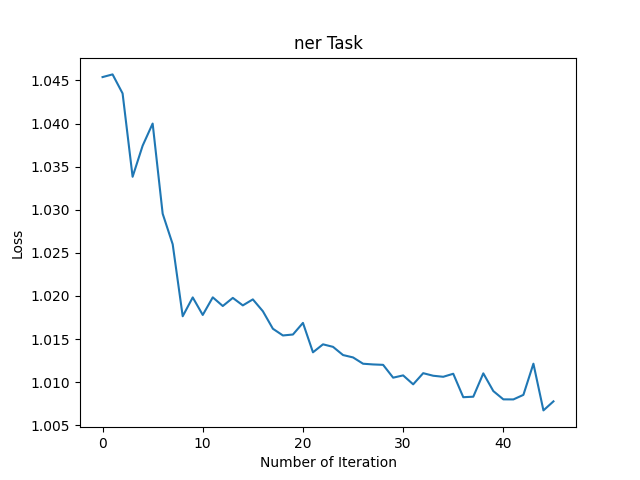
**Number of filters** – 30

**Window size** – 3

**Accuracy**: 84.6 in dev



**Loss:**

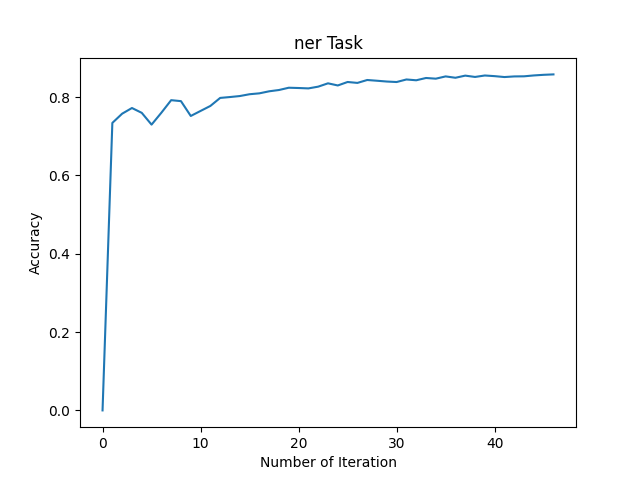


**Experiment 2:**

**Number of filters** – 50

**Window size** – 5

**Accuracy**: 85.58 in dev



**Loss**:

A picture containing text, diagram, line, plot

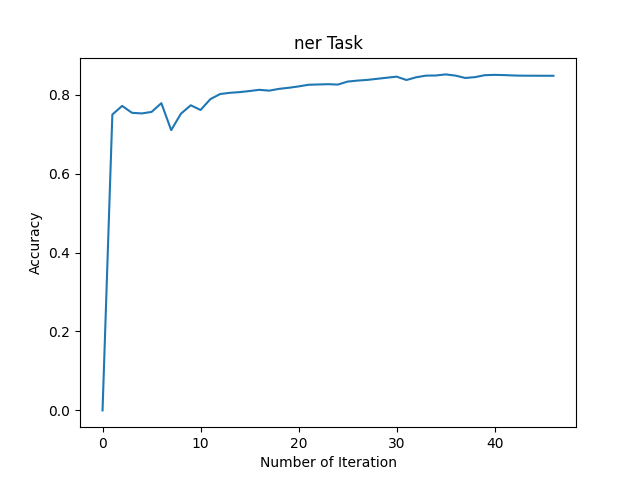
Description automatically generated

**Experiment 3:**

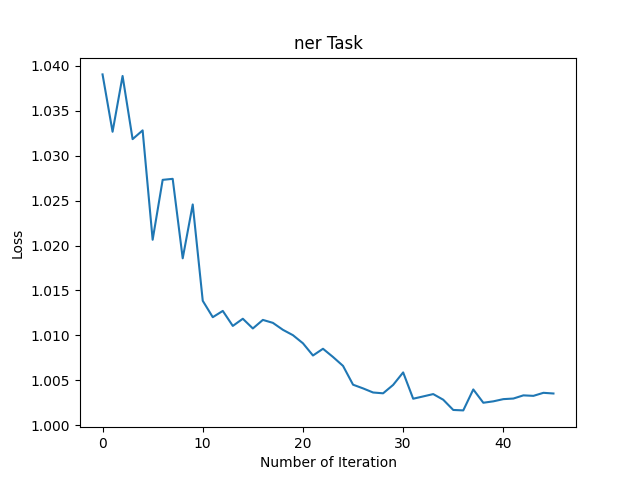
**Number of filters** – 30

**Window size** – 5

**Accuracy**: 84.78 in dev



**Loss:**

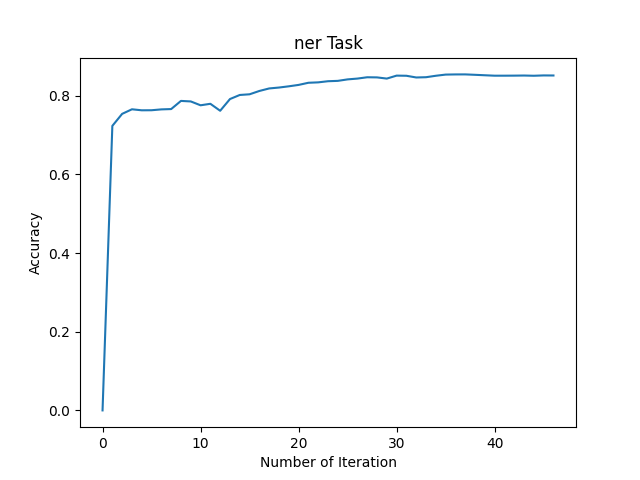


**Experiment 4:**

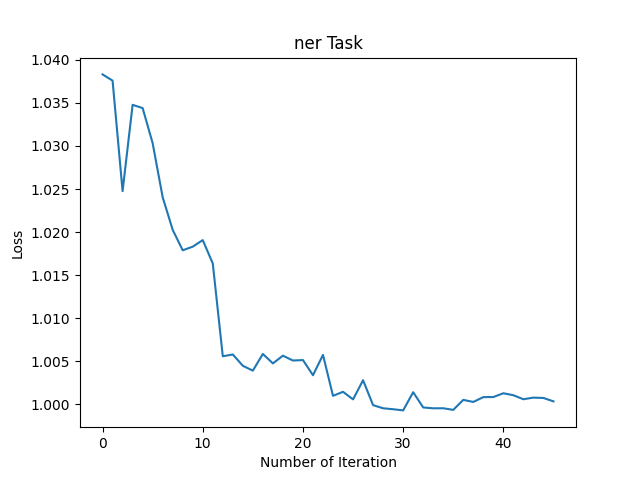
**Number of filters** – 50

**Window size** – 3

**Accuracy**: 85.11 in dev

****

**Loss:**

****

**Extracting Meaningful features from convolution layer**

We conducted three distinct experiments aimed at analyzing the convolution filters in our model. Our primary focus was on understanding the behavior and characteristics of these filters.

In the first experiment, we extracted the triplets with the highest mean score for each of the 30 filters in the convolutional layer. By examining these triplets, we sought to determine if different filters were capturing specific patterns or features. This analysis provided insights into the types of information that each filter was sensitive to.

In the second experiment, we investigated the relationship between labels, filters, and triplets. For each label and filter combination, we examined the number of times each triplet achieved the highest score. This analysis allowed us to identify which triplets were consistently associated with high scores across different labels and filters, indicating their importance in the model's decision-making process.

In the third experiment, we focused on the individual triplets that received the highest scores. For each label, we analyzed the triplets that consistently obtained the highest score, irrespective of the filter. This investigation provided valuable insights into the specific triplets that played a crucial role in determining the model's predictions.

**Argmax by Filter, Label and Triplet – Experiment 1:**

In this part, we saved for each label the triplets that gave the highest scores

LOC – a bunch of filters have those kinds of triplets:

**Filter 6:**

1. “se”: “se” or “south-east” in phrases like “south-east Asia” or “south-east region.”
2. “le”: “le” or “east” in phrases like “east coast” or “eastern region.”
3. “ca”: “ca” or “California” in phrases like “California state” or “cities in California.”
4. “we”: This filter could capture the location feature “we” or “west” in phrases like “west coast” or “western region.”
5. “di”: “di” or “district” in phrases like “financial district” or “downtown district.”
6. “ch”: This filter likely captures the location feature “ch” or “city” in phrases like “city center” or “urban city.”
7. “ha”: “ha” or “harbor” in phrases like “harbor area” or “port harbor.”
8. “be”: “be” or “beach” in phrases like “beachfront” or “coastal beach.”
9. “ba”: “ba” or “bay” in phrases like “bay area” or “bayfront.”
10. “pr”: “pr” or “province” in phrases like “province of Alberta” or “provincial capital.”
11. “de”: “de” or “department” in phrases like “department of transportation” or “government department.”

**Filter 22 – Part of the triplets related to directions:**

1. "est": "west" or "east" in phrases like "west coast" or "eastern region."
2. "rom": "from"
3. "nor": "north" in phrases like "northern region" or "northbound."

**ChatGPT helped us to understand this relationship ☺**

**Average filter score for each triplet in train set– Experiment 2:**

**Triplets with Numbers:**

Filter 21: ['-3-', '-2-', '-i-', '-5-', '-8-', '-7-', '-6-', '03-', '02-', '-4-']

Filter 22: [‘090’, ‘990’, ‘-90’, ‘0.0’, ‘190’, ‘040’, ‘490’, ‘.90’, ‘ 90’, ‘/90’]

**Triplets with OO:**

Filter 13: ['ooo', 'oos', 'ook', 'oo ', 'oon', 'ooh', 'ooz', 'ooi', 'oof', 'soo']

Filter 1: ['poo', 'roo', 'hoo', 'soo', 'voo', 'boo', 'loo', ' oo', 'joo', 'moo']

**Numbers with Dash:**

Filter 27: ['-0-', '---', '70-', '7--', '30-', '60-', '80-', '20-', '50-', ' 0-']

**Dash in the middle:**

Filter 3: ['o-y', 'o-p', '9-3', 'o-t', 'o-l', 's--', 'o-m', '9-2', 'o-r', 'o-a']

**Part of Speech**

**Experiment 1:**

**Number of filters** – 50

**Window size** – 3

**Accuracy**: 85.11 in dev