Assignment 2 - Knowledge Graph Population - Report

The scope of this assignment is to create an LLM-based relation classifier, which takes as input a given sentence and two entities referring to it, and determines which relation between them is true based on the sentence. The available relations that can be derived from sentences are as follows.

Relations

- **cities_of_residence**: relates a person to the cities they currently live or have lived in the past
- **employee_of**: relates a person to the organizations they are currently employees of or have been in the past
- **schools_attended**: relates a person to the schools they are currently attending or have attended in the past.
- **spouse**: relates a person to the persons they are currently married to or have been married to in the past

However, if a given sentence does not express any of the above relations between the entities, the classifier should return the word 'unknown' as it is not clear what is the real relationship between them and cannot get an accurate conclusion.

The assignment consists of 2 tasks, where the first task asks to build the above classifier and use a provided dataset to evaluate the precision and recall metrics for each relation, while the second task asks to create a small dataset with the same format as the provided evaluation dataset of task 1 (sentence-entities-relation), which tries to evaluate and realize how well the existing classifier handles the phenomena of 'uncertainty' and 'advice/wish'.

In the next two chapters we will look at the steps and design decisions made in both tasks to achieve the desired results.

Task 1 - LLM-Based Relation Classifier

In order to create the classifier, I decided to use ChatGPT LLM and in particular its gpt-4o-mini model. The promt design in the ChatGPT API consists of 2 important roles which are the 'system' and 'user'. The 'system' role allows you to specify how the model should behave and answer the given questions while the 'user' role equates to the questions the user asks the model.

I tried experimenting with variations of promts to see which works best for our work. Below are all the promts I used until I found the one that fits our specific task exactly. In all the

following promts the content of the 'user' role is always the same and it is:

```
Given the sentence: 'sentence', what is the relationship between
the entities 'entity_1' and 'entity_2'?
```

Beacuse, in the following promts we will calculate the precision and recall of the relationships, here is the formulas of them:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

First Promt

In the first attempt I gave to the 'system' role only the context that it is a system that identifies relationships between entities in a sentence and that the available relationships are spouse, schools_attended, employee_of and cities_of_residence. I also defined some specific instructions which are to return 'unknown' if none of the relationships are valid in the sentence and also to return in each case only a word representing the name of the relationship.

In particular, the content of the 'system' role was:

```
You are a system that identifies relationships between two entities
in a sentence. The possible relationships are 'spouse',
'schools_attended', 'employee_of' and 'cities_of_residence'.
Important Instructions:
    - If none of the above relationships are expressed in the
sentence, return 'unknown'.
    - Return only a single word that represents the relationship
(e.g., 'spouse', 'employee_of', or 'unknown'). Do not include any
extra text or explanations.
```

The precision and recall of each relationship based on this prompt is as follows:

Relation	Precision	Recall
spouse	0.21	1.00
employee_of	0.12	1.00
cities_of_residence	0.18	1.00
schools_attended	0.37	1.00
unknown	1.00	0.38

After using this prompt, I noticed that the precision of the relationships were very low because the system cannot understand in many cases the correct relationship between the entities in the sentences. In particular, the problem was when the actual relation was 'unknown' and the system could not understand it and returned another relation. One

possible reason for this was that it did not explicitly know what each relation means for our task. Because of this, I decided to embed the descriptions of each relation in the promt.

Second Promt

In the second promt, i incorporated the descriptions of each relationship into the prompt. Thus, the content of the 'system' role became:

```
You are a system that identifies relationships between two entities
in a sentence. The possible relationships are:
            1. 'spouse': relates a person to the persons they are
currently married to or have been married to in the past.
            2. 'schools_attended': relates a person to the schools
they are currently attending or have attended in the past.
            3. 'employee_of': relates a person to the organizations
they are currently employees of or have been in the past.
            4. 'cities_of_residence': relates a person to the
cities they currently live or have lived in the past.
Important Instructions:
    - If none of the above relationships are expressed in the
sentence, return 'unknown'.
    - Return only a single word that represents the relationship
(e.g., 'spouse', 'employee_of', or 'unknown'). Do not include any
extra text or explanations.
```

The precision and recall of each relationship based on this prompt is as follows:

Relation	Precision	Recall
spouse	0.20	1.00
employee_of	0.12	1.00
cities_of_residence	0.12	1.00
schools_attended	0.36	1.00
unknown	1.00	0.28

Despite the reason, i added this additional information to the prompt, it didn't have much more precision and accuracy (e.g. precision(cities_of_residence) = 0.12, precision(spouse) = 0.20 etc.) and again the system needed more guidance to figure out how to classify each case and especially when the actual relation was 'unknown'. So, the next thing i added to the prompt was some classification examples for each relationship to give the classifier an idea of how and when a relationship is valid between two entities.

Third Promt

In the third promt, I incorporated some examples into the prompt. Thus, the content of the 'system' role became:

```
You are a system that identifies relationships between two entities
in a sentence. The possible relationships are:
    1. 'spouse': relates a person to the persons they are currently
married to or have been married to in the past.
    2. 'schools_attended': relates a person to the schools they are
currently attending or have attended in the past.
    3. 'employee_of': relates a person to the organizations they
are currently employees of or have been in the past.
    4. 'cities_of_residence': relates a person to the cities they
currently live or have lived in the past.
Important Instructions:
    - If none of the above relationships are expressed in the
sentence, return 'unknown'.
    - Return only a single word that represents the relationship
(e.g., 'spouse', 'employee_of', or 'unknown'). Do not include any
extra text or explanations.
Below are some examples for how to handle these cases:
    Example 1:
    Sentence: 'John and Jane were married two weeks ago.'
    Subject: John
    Object: Jane
    Return: spouse

    Example 2:
    Sentence: 'Alice graduated from MIT in 2005.'
    Subject: Alice
    Object: MIT
    Return: schools_attended

    Example 3:
    Sentence: 'Mark lives in San Francisco and also spent several
years in New York.'
    Subject: Mark
    Object: San Francisco
    Return: cities_of_residence

    Example 4:
    Sentence: 'Maria works for IBM as a software engineer.'
    Subject: Maria
    Object: IBM
    Return: employee_of
```

The precision and recall of each relationship based on the last prompt is as follows:

Relation	Precision	Recall
spouse	0.28	1.00
employee_of	0.13	1.00
cities_of_residence	0.15	1.00
schools_attended	0.38	1.00

Relation	Precision	Recall
unknown	1.00	0.42

After this addition, the results were better and the precision of the relationships became greater, but not quite as effective. I noticed that the classifier could not understand cases where we have "negations" or the sentences did not express their content with certainty and the result was uncertain. Because of this, I decided to give some additional instructions to the system to only consider sentences that express something accurately, otherwise it would return 'unknown' and if a relation is expressed as a negative in the sentence it would still return 'unknown'.

Below there are some examples from the initial dataset which have "negations" or uncertainty:

1. Jane has never been John's wife.
2. It's possible that John is employed by Apple.
3. Jane is not a wife of John.
4. It's possible that John lived in Palo Alto.

Fourth Promt

In the fourth attempt, in order to teach the system to return 'unknown' when a relation is expressed as a negative in a sentence or the content of the sentence is uncertain, I decided to give some instructions about this and also to give an example for each case.

So, the content of the 'system' role became:

```
You are a system that identifies relationships between two entities
in a sentence. The possible relationships are:
    1. 'spouse': relates a person to the persons they are currently
married to or have been married to in the past.
    2. 'schools_attended': relates a person to the schools they are
currently attending or have attended in the past.
    3. 'employee_of': relates a person to the organizations they
are currently employees of or have been in the past.
    4. 'cities_of_residence': relates a person to the cities they
currently live or have lived in the past.
Important Instructions:
    - If the relationship is negated in the sentence (e.g., 'John
is not married to Jane'), return 'unknown'.
    - If the relationship is expressed with uncertainty or
probability (e.g., 'John may be employed by Google' or 'John could
have lived in San Francisco'), return 'unknown'.
    - If none of the above relationships are expressed in the
sentence, return 'unknown'.
    - Return only a single word that represents the relationship
(e.g., 'spouse', 'employee_of', or 'unknown'). Do not include any
extra text or explanations.
```

```
Below are some examples for how to handle these cases:
    Example 1:
    Sentence: 'John and Jane were married two weeks ago.'
    Subject: John
    Object: Jane
    Return: spouse

    Example 2:
    Sentence: 'Alice graduated from MIT in 2005.'
    Subject: Alice
    Object: MIT
    Return: schools_attended

    Example 3:
    Sentence: 'Mark lives in San Francisco and also spent several
years in New York.'
    Subject: Mark
    Object: San Francisco
    Return: cities_of_residence

    Example 4:
    Sentence: 'Maria works for IBM as a software engineer.'
    Subject: Maria
    Object: IBM
    Return: employee_of

    Example 5 (Negation):
    Sentence: 'John is not married to Jane.'
    Subject: John
    Object: Jane
    Return: unknown

    Example 6 (Uncertainty):
    Sentence: 'John may work at Google.'
    Subject: John
    Object: Google
    Return: unknown
```

The precision and recall of each relationship based on the last prompt is as follows:

Relation	Precision	Recall
spouse	0.55	1.00
employee_of	0.89	1.00
cities_of_residence	0.82	0.90
schools_attended	0.93	1.00
unknown	1.00	0.95

After this addition, the results were very good and the only thing I noticed that was wrong was the sentences expressing something in the future. It seems that the system, despite the descriptions of the relations referring to considering only the past or current time, also takes the future into account. The correct behaviour is that when a relation is expressed for the future then the system should not return the relation but the word 'unknown'.

Below there are some sentences from the initial dataset which express something in the future:

1. Jane will be John's wife.
2. John and Jane are getting married.
3. John and Jane's wedding is coming up.

Fifth and Last Promt

In the last prompt, to avoid cases where a relationship is expressed for the future, I simply added the comment "Not in the future" to the description of each relationship.

So the content of the 'system' role in the prompt is:

```
You are a system that identifies relationships between two entities
in a sentence. The possible relationships are:
    1. 'spouse': relates a person to the persons they are currently
married to or have been married to in the past. Not in the future.
    2. 'schools_attended': relates a person to the schools they are
currently attending or have attended in the past. Not in the
future.
    3. 'employee_of': relates a person to the organizations they
are currently employees of or have been in the past. Not in the
future.
    4. 'cities_of_residence': relates a person to the cities they
currently live or have lived in the past. Not in the future.

Important Instructions:
    - If the relationship is negated in the sentence (e.g., 'John
is not married to Jane'), return 'unknown'.
    - If the relationship is expressed with uncertainty or
probability (e.g., 'John may be employed by Google' or 'John could
have lived in San Francisco'), return 'unknown'.
    - If none of the above relationships are expressed in the
sentence, return 'unknown'.
    - Return only a single word that represents the relationship
(e.g., 'spouse', 'employee_of', or 'unknown'). Do not include any
extra text or explanations.
Below are some examples for how to handle these cases:
    Example 1:
    Sentence: 'John and Jane were married two weeks ago.'
    Subject: John
    Object: Jane
    Return: spouse
```

```
    Example 2:
    Sentence: 'Alice graduated from MIT in 2005.'
    Subject: Alice
    Object: MIT
    Return: schools_attended

    Example 3:
    Sentence: 'Mark lives in San Francisco and also spent several
years in New York.'
    Subject: Mark
    Object: San Francisco
    Return: cities_of_residence

    Example 4:
    Sentence: 'Maria works for IBM as a software engineer.'
    Subject: Maria
    Object: IBM
    Return: employee_of

    Example 5 (Negation):
    Sentence: 'John is not married to Jane.'
    Subject: John
    Object: Jane
    Return: unknown

    Example 6 (Uncertainty):
    Sentence: 'John may work at Google.'
    Subject: John
    Object: Google
    Return: unknown
```

To mention that in the final classifier in the jupyter notebook, i use the fifth and last prompt.

The precision and recall of each relationship based on the last prompt is as follows:

Relation	Precision	Recall
spouse	0.92	1.00
employee_of	1.00	1.00
cities_of_residence	1.00	0.90
schools_attended	1.00	0.93
unknown	0.99	1.00

Just to mention, that in all promts the recall of the spouse, employee_of, cities_of_residence and schools_attended relations, is very good which means that the classifier is able to determine the correct relation when the actual one is not the 'unknown'.

Task 2 - Uncertainty and Advice/Wish Phenomena

Uncertainty

This is when the input sentence expresses the possibility or probability that a relation is true between two entities, without being sure that it is (e.g.: 'It is possible that John is an employee of Apple.' or 'It's possible that John lived in Palo Alto.')

Advice/Wish

This is when the input sentence expresses the wish or suggestion that a relation is true, without that necessarily being the case (e.g. 'It would be nice if John married Jane').

For this task, in order to evaluate how well the classifier satisfies the 'uncertainty' and 'advice/wish' phenomena, I created a new dataset containing 80 records in the same format as the dataset of Task 1, where for each relation i constructed 20 sentences expressing the relation, but in a 'probability'(10 sentences) or 'advice/wish'(10 sentences) manner. The correct relation of the entities included in these sentences must be 'unknown', because none of the other relations are valid.

In order to create the appropriate dataset, i use the ChatGPT giving it the promt as follows:

```
I have a classifier that reads a sentence and tries to understand
the relationship between two specific entities based on the
sentence. The possible relationships are:

1. "spouse": relates a person to the persons they are currently
married to or have been married to in the past.
2. 'schools_attended': relates a person to the schools they are
currently attending or have attended in the past.
3. 'employee_of':  relates a person to the organizations they are
currently employees of or have been in the past.
4. "cities_of_residence": relates a person to the cities they
currently live or have lived in the past.

If none of the above relationships are expressed in the sentence
then the classifier should return unknown. I want you to help me
create a small dataset with sentences, entities(subject,object) and
relations in order to evaluate my classifier on the following
phenomena:

Uncertainty: This is when the input sentence expresses the
possibility or probability that
a relation is true between two entities, without being sure that it
is (e.g.: “It is possible
that John is an employee of Apple.” or “It's possible that John
lived in Palo Alto.”), In
such cases, the classifier should not suggest that this relation
```

```
holds and returns unknown.

Advice/Wish: This is when the input sentence expresses the wish or
suggestion that a
relation is true, without that necessarily being the case (e.g. “It
would be nice if John
married Jane”). In such cases, the classifier should not suggest
that this relation
holds and returns unknown.

I want 10 sentences for each relation and phenomenon. The output
should be a table with columns setence, subject, object and
relation.
```

The following table shows the precision and recall of the relations after running the classifier on the new dataset.

Relation	Precision	Recall
spouse	0.00	0.00
employee_of	0.00	0.00
cities_of_residence	0.00	0.00
schools_attended	0.00	0.00
unknown	1.00	1.00

From the above table, we see that the classifier fully satisfies both phenomena, as it returns the 'unknown' relation for all sentences which is the true one. This happens because in task 1, i had already taken into account the 'uncertainty' or 'probability' factors as there are cases with them and i had to handle them in order to enhance the precision of the relationships.