Personal Information

Name: **Ioana Mazilu**

StudentID: **14642484**

Email: **ioana.mazilu@student.uva.nl**

Submitted on: **18.03.2024**

Data Context

The aim of my research project is to test the ability of base and fine-tuned language models (small - 7B/13B and large 1.76T parameters) to generate Python scripts that can perform calculations and comparisons and return the correct classification label for a pair of premise and hypothesis. This task is known as Quantitative Natural Language Inference (QNLI), and it is derived from the NLI task, but focuses only on sentences with quantitative information.

The dataset I will be using is called EQUATE, which is a benchmark dataset for QNLI, introduced by Ravichander et al. [1]. It consists of 5 sub-datasets, with various characteristics. Three of the datasets are of natural-language source (NewsNLI - from news articles, RedditNLI - from financial headlines on Reddit, and RTE_Quant - from a dataset for numerical reasoning). The other 2 datasets are of synthetic nature (AWPNLI - derived from math word problems, and StressTest, derived from algebra word problems). The datasets consist of a premise, hypothesis and label, which is one of entailment, contradiction or neutral. There are other features as well in the datasets, created from processing of the premise/hypothesis features. However, for my project only the former-named columns are relevant. For the EDA, we will also take a look at some of the other columns.

The code related to this project is stored at https://github.com/IoanaMazilu/msc_qnli

References

[1] References Ravichander, A., Naik, A., Rosé, C. P., & Hovy, E. H. (2019). EQUATE: A Benchmark Evaluation Framework for Quantitative Reasoning in Natural Language Inference. CoRR, abs/1901.03735. Retrieved from http://arxiv.org/abs/1901.03735

Data description

The EDA reveals that the 5 datasets contain different combinations of labels. For instance, StressTest and RedditNLI have samples from all 3 categories, while the other datasets have samples from the entailment category and the second label is either neutral or contradiction. The RedditNLI dataset is highly imbalanced. The frequency of each label in each dataset is as follows:

- RedditNLI: entailment (57.6%), neutral (34%), contradiction (7.6%)
- NewsNLI: entailment (50.7%), neutral (49.3%)
- RTE_Quant: entailment (42.2%), neutral (57.8%)
- AWPNLI: entailment (50%), contradiction (50%)
- StressTest: entailment (33.3%), neutral (33.3%), contradiction (33.3%)

At the level of EQUATE, there is also an imbalance, as StressTest has over 7K samples, while all other datasets have less than 1K samples, and RTE_Quant even less than 200:

- RedditNLI: 2.58%
- NewsNLI: 9.98%
- RTE_Quant: 1.71%
- AWPNLI: 7.44%
- StressTest: 78.29%

In terms of studying the quality of the data and if there are samples which need to be discarded (since correcting them is not an option in our case), we look at duplicates and sentences that do not contain quantitative information (which is a requirement for the QNLI task). We inspect the duplicates, and find that StressTest has the most duplicates, namely 649, or 8.54% of all the samples, while all other datasets have at most 5 duplicates. We also identify samples where either the premise or hypothesis potentially do not contain quantitative information. Manual inspection of these samples (as they are few) reveals which samples can be kept and which must be discarded. One interesting insight is about the RTE_Quant dataset, for which annotator labels are provided. In more than half of instances, there is a disagreement between the annotators regarding the correct label. This could be an indicator of the higher difficulty of samples in this set, and it is worth keeping in mind during the data generation and evaluation parts.

By looking at the length of the premise and hypothesis, we observe that some datasets have almost equal-length premises and

hypotheses. We assume this is because the focus is on a direct comparison of the quantities in the inputs and/or identifying if the hypothesis is not related to the premise (which is usually the neutral class). Conversely, for the sets where the hypothesis is much shorter than the premise, we observe that the former is usually a shorter, rephrased version of the latter (2 or 3 times shorter). We also find that AWPNLI involves calculations using the quantities in the premise, obtaining a final value which must be compared to the one in the hypothesis.

Using word-clouds to inspect the most frequent unigrams at both the premise and hypothesis level also reveals some interesting insights about the topics covered in the datasets and what types of quantities can be found inside them. For instance, AWPNLI (derived from math word problems) contains a lot of simple nouns (i.e., orange, apple, dimes, books etc.) and verbs indicating either addition or subtraction (picked, left, bought, needed). This also suggests that samples in this dataset will require calculations before a comparison can be made to infer the QNLI label.

Finally, we analyze the amount of samples in each set that contain textual quantifiers. We create a (non-exhaustive) list of common quantifiers. We find that the synthetic datasets (math-based) and the NewsNLI set contain the most samples with quantifiers. While AWPNLI contains only 4 unique quantifiers, the other 2 datasets (StressTest and NewsNLI) contain a more diverse set of quantifiers.

In [1]:
```
import jsonlines
import os
import re

import numpy as np
import pandas as pd
```

In [2]:
```
# go back 2 directories from the cwd
root_path = os.path.dirname(os.path.dirname(os.getcwd()))
data_directory_path = os.path.join(root_path, "data", "equate")
```

Data Loading

In [3]:
```
def read_data(filename):
    ''' Reads a jsonl file
    :param filename: file to be read
    :return: list of NLI samples
    '''
    print(f"###############\nData file: {filename.split('/')[-1]}")
    samples = []
    with jsonlines.open(os.path.join(data_directory_path, filename)) as reader:
        for obj in reader:
            samples.append(obj)
    assert len(samples) > 0
    labels = set([sample['gold_label'] for sample in samples]) # unique labels in the dataset
    samples_df = pd.DataFrame(samples)
    samples_df["sample_index"] = samples_df.index
    samples_df = samples_df.rename(columns={"sentence1": "premise", "sentence2": "hypothesis", "gold_label": "la
    print(f"Dataset features: {samples_df.columns}")
    return samples_df, labels
```

In [4]:
```
datasets = ["AWPNLI.jsonl", "NewsNLI.jsonl", "RedditNLI.jsonl", "RTE_Quant.jsonl", "StressTest.jsonl"]
datasets = [os.path.join(data_directory_path, dataset) for dataset in datasets]

awp, awp_labels = read_data(datasets[0])
news, news_labels = read_data(datasets[1])
reddit, reddit_labels = read_data(datasets[2])
rte, rte_labels = read_data(datasets[3])
stress, stress_labels = read_data(datasets[4])
```

```
###############
Data file: AWPNLI.jsonl
Dataset features: Index(['sentence2_tokens', 'sentence1_dep_parse', 'sentence1_binary_parse',
       'sentence2_binary_parse', 'sentence1_syntax_parse', 'premise',
       'hypothesis', 'sentence2_syntax_parse', 'hypothesis_pos',
       'sentence2_dep_parse', 'label', 'premise_pos', 'sentence1_tokens',
       'sample_index'],
      dtype='object')
###############
Data file: NewsNLI.jsonl
Dataset features: Index(['sentence2_tokens', 'annotator_labels', 'sentence1_tokens',
       'sentence1_dep_parse', 'sentence2_syntax_parse', 'Phenomena',
       'sentence1_binary_parse', 'Hard', 'sentence2_parse',
       'sentence2_binary_parse', 'sentence1_syntax_parse', 'premise',
       'hypothesis', 'sentence1_parse', 'genre', 'hypothesis_pos',
       'sentence2_dep_parse', 'label', 'premise_pos', 'PairID',
       'sample_index'],
      dtype='object')
###############
Data file: RedditNLI.jsonl
Dataset features: Index(['sentence2_tokens', 'sentence1_tokens', 'sentence1_dep_parse',
       'sentence2_syntax_parse', 'sentence1_binary_parse', 'sentence2_parse',
       'sentence2_binary_parse', 'sentence1_syntax_parse', 'premise',
       'hypothesis', 'sentence1_parse', 'genre', 'hypothesis_pos',
       'sentence2_dep_parse', 'label', 'premise_pos', 'PairID',
       'sample_index'],
      dtype='object')
###############
Data file: RTE_Quant.jsonl
Dataset features: Index(['sentence2_tokens', 'annotator_labels', 'sentence1_dep_parse',
       'sentence1_binary_parse', 'sentence2_binary_parse',
       'sentence1_syntax_parse', 'premise', 'hypothesis',
       'sentence2_syntax_parse', 'genre', 'hypothesis_pos',
       'sentence2_dep_parse', 'label', 'premise_pos', 'sentence1_tokens',
       'sample_index'],
      dtype='object')
###############
Data file: StressTest.jsonl
Dataset features: Index(['sentence2_tokens', 'sentence1_dep_parse', 'sentence1_binary_parse',
       'sentence2_binary_parse', 'sentence1_syntax_parse', 'premise',
       'hypothesis', 'sentence2_syntax_parse', 'hypothesis_pos',
       'sentence2_dep_parse', 'label', 'premise_pos', 'sentence1_tokens',
       'sample_index'],
      dtype='object')
```

We notice some of the datasets have unique features (i.e. RTE_Quant has a 'genre' feature, NewsNLI has a 'genre', 'Hard', 'Phenomena' and 'annotator_labels'. We will inspect these features later to find out what they represent.

Explanation of the columns, relevant for the EDA:

- premise: the premise
- hypothesis: the hypothesis
- label: the NLI classification label (entailment/neutral/contradiction)
- premise_pos, hypothesis_pos: For each sentence in the premise/hypothesis, a list is extracted of the role of each word in the sentence. Example: For the sentence "15.0 pizzas were served today", the following list of word roles is extracted: [["CD", "NNS", "VBD", "VBN", "NN"]]. We observe that "CD" represents quantities.

In [5]:
```
awp.head(5)
```

Out[5]:

	sentence2_tokens	sentence1_dep_parse	sentence1_binary_parse	sentence2_binary_parse	sentence1_syntax_parse	premise	h
0	[[sam, has, 16.0, dimes, now]]	[[{'dep': 'ROOT', 'dependent': 11, 'governorGl...	Sam had 9.0 dimes in his bank and his dad gav...	Sam has 16.0 dimes now	[(ROOT\n (NP\n (S\n (S\n (NP (...	Sam had 9.0 dimes in his bank and his dad gav...	
1	[[sam, has, 17.0, dimes, now]]	[[{'dep': 'ROOT', 'dependent': 11, 'governorGl...	Sam had 9.0 dimes in his bank and his dad gav...	Sam has 17.0 dimes now	[(ROOT\n (NP\n (S\n (S\n (NP (...	Sam had 9.0 dimes in his bank and his dad gav...	
2	[[15.0, pizzas, were, served, today]]	[[{'dep': 'ROOT', 'dependent': 3, 'governorGlo...	A restaurant served 9.0 pizzas during lunch an...	15.0 pizzas were served today	[(ROOT\n (S\n (NP (DT a) (NN restaurant))\...	A restaurant served 9.0 pizzas during lunch an...	1
3	[[17.0, pizzas, were, served, today]]	[[{'dep': 'ROOT', 'dependent': 3, 'governorGlo...	A restaurant served 9.0 pizzas during lunch an...	17.0 pizzas were served today	[(ROOT\n (S\n (NP (DT a) (NN restaurant))\...	A restaurant served 9.0 pizzas during lunch an...	1
4	[[5.0, pencils, are, now, there, in, total]]	[[{'dep': 'ROOT', 'dependent': 2, 'governorGlo...	There are 2.0 pencils in the drawer and Tim p...	5.0 pencils are now there in total	[(ROOT\n (S\n (NP (EX there))\n (VP (VB...	There are 2.0 pencils in the drawer and Tim p...	

Analysis of sample counts per sub-dataset in EQUATE and per language type (natural/synthetic)

How many samples are in each dataset?

In [6]:
```
print(len(awp), len(news), len(reddit), len(rte), len(stress))
```

```
722 968 250 166 7596
```

What fraction of the EQUATE benchmark each sub-dataset is?

In [7]:
```
equate_size = len(awp) + len(news) + len(reddit) + len(rte) + len(stress)
print(round(len(awp)*100/equate_size, 2), round(len(news)*100/equate_size, 2), round(len(reddit)*100/equate_size
```

```
7.44 9.98 2.58 1.71 78.29
```

The datasets are of 2 types:

- based on natural, every-day language, scraped from sources like Reddit (RedditNLI), news articles (NewsNLI) and a dataset of quantitative problems (RTE_Quant);
- based on synthetic language, created from Math World Problems (MWPs) (StressTest, AWPNLI). Let's inspect how many samples of each category we have

In [8]:
```
natural_language_samples = news.shape[0] + reddit.shape[0] + rte.shape[0]
synthetic_language_samples = stress.shape[0] + awp.shape[0]
total_samples = natural_language_samples + synthetic_language_samples
print(f"Natural-language samples: {natural_language_samples} ({round((natural_language_samples/total_samples)*1
print(f"Synthetic-language samples: {synthetic_language_samples} ({round((synthetic_language_samples/total_samp
```

```
Natural-language samples: 1384 (14.27% of all samples in EQUATE)
Synthetic-language samples: 8318 (85.73% of all samples in EQUATE)
```

We notice a significant imbalance at the EQUATE dataset level between natural language samples and synthetic language samples, with a large ratio of the total samples being of synthetic nature, and more specifically from the StressTest dataset.

Analysis of the sentences which form the premises and hypotheses.

UTIL FUNCTIONS

In [9]:
```
def clean_text(df: pd.DataFrame, column_name: str):
    df[column_name] = df[column_name].apply(lambda str_value: re.sub(r'\s+', ' ', str_value.replace("\n", "")).
```

In [10]:
```
def count_words_in_string(df: pd.DataFrame, column_name: str):
    df[f"{column_name}_word_cnt"] = df[column_name].apply(lambda str_value: len(str_value.split(" ")))
```

In [11]:
```
def count_chars_in_string(df: pd.DataFrame, column_name: str):
    df[f"{column_name}_char_cnt"] = df[column_name].apply(lambda str_value: len(str_value))
```

In [12]:
```
def quantities_in_sentence(df: pd.DataFrame, column_name: str):
    df[f"{column_name}_quantities_cnt"] = df[f"{column_name}_pos"].apply(lambda entities: np.sum([1 for i in ra
```

In [13]:
```
def unique_annotator_labels(df: pd.DataFrame):
    df["annotator_unique_labels"] = df["annotator_labels"].apply(lambda labels_array: len(set(labels_array)))
```

In [14]:
```
def sentence_insights(dataset_df):
    clean_text(dataset_df, "premise")
    clean_text(dataset_df, "hypothesis")
    count_words_in_string(dataset_df, "premise")
    count_words_in_string(dataset_df, "hypothesis")
    count_chars_in_string(dataset_df, "premise")
    count_chars_in_string(dataset_df, "hypothesis")
    quantities_in_sentence(dataset_df, "premise")
    quantities_in_sentence(dataset_df, "hypothesis")
    if "annotator_labels" in dataset_df.columns:
        unique_annotator_labels(dataset_df)
    else:
        print("There is no data on the annotator labels for this dataset.")
```

AWPNLI dataset

In [15]:
```
awp.head()
```

Out[15]:

	sentence2_tokens	sentence1_dep_parse	sentence1_binary_parse	sentence2_binary_parse	sentence1_syntax_parse	premise	h
0	[[sam, has, 16.0, dimes, now]]	[[{'dep': 'ROOT', 'dependent': 11, 'governorGl...	Sam had 9.0 dimes in his bank and his dad gav...	Sam has 16.0 dimes now	[(ROOT\n (NP\n (S\n (S\n (NP (...	Sam had 9.0 dimes in his bank and his dad gav...	
1	[[sam, has, 17.0, dimes, now]]	[[{'dep': 'ROOT', 'dependent': 11, 'governorGl...	Sam had 9.0 dimes in his bank and his dad gav...	Sam has 17.0 dimes now	[(ROOT\n (NP\n (S\n (S\n (NP (...	Sam had 9.0 dimes in his bank and his dad gav...	
2	[[15.0, pizzas, were, served, today]]	[[{'dep': 'ROOT', 'dependent': 3, 'governorGlo...	A restaurant served 9.0 pizzas during lunch an...	15.0 pizzas were served today	[(ROOT\n (S\n (NP (DT a) (NN restaurant))\...	A restaurant served 9.0 pizzas during lunch an...	1
3	[[17.0, pizzas, were, served, today]]	[[{'dep': 'ROOT', 'dependent': 3, 'governorGlo...	A restaurant served 9.0 pizzas during lunch an...	17.0 pizzas were served today	[(ROOT\n (S\n (NP (DT a) (NN restaurant))\...	A restaurant served 9.0 pizzas during lunch an...	1
4	[[5.0, pencils, are, now, there, in, total]]	[[{'dep': 'ROOT', 'dependent': 2, 'governorGlo...	There are 2.0 pencils in the drawer and Tim p...	5.0 pencils are now there in total	[(ROOT\n (S\n (NP (EX there))\n (VP (VB...	There are 2.0 pencils in the drawer and Tim p...	

In [16]:
```
awp['label'].value_counts(normalize=True)
```

Out[16]:
```
label
entailment       0.5
contradiction    0.5
Name: proportion, dtype: float64
```

The distribution of samples across the 2 labels is balanced

In [17]:
```
sentence_insights(awp)

awp[["premise", "hypothesis", "label", "premise_word_cnt", "hypothesis_word_cnt", "premise_char_cnt", "hypothes
```

```
There is no data on the annotator labels for this dataset.
```

Out[17]:

	premise	hypothesis	label	premise_word_cnt	hypothesis_word_cnt	premise_char_cnt	hypothesis_char_cnt
0	Sam had 9.0 dimes in his bank and his dad gave...	Sam has 16.0 dimes now	entailment	14	5	60	22
1	Sam had 9.0 dimes in his bank and his dad gave...	Sam has 17.0 dimes now	contradiction	14	5	60	22
2	A restaurant served 9.0 pizzas during lunch an...	15.0 pizzas were served today	entailment	13	5	73	29
3	A restaurant served 9.0 pizzas during lunch an...	17.0 pizzas were served today	contradiction	13	5	73	29
4	There are 2.0 pencils in the drawer and Tim pl...	5.0 pencils are now there in total	entailment	15	7	76	34

In [18]:
```
awp[["premise_word_cnt", "hypothesis_word_cnt", "premise_char_cnt", "hypothesis_char_cnt"]].aggregate(["mean",
```

Out[18]:

	premise_word_cnt	hypothesis_word_cnt	premise_char_cnt	hypothesis_char_cnt
mean	16.236842	6.450139	84.925208	33.279778
std	5.694548	1.919570	32.652246	10.538450

We notice that the premises are larger than the hypotheses, on average, by at least 2 times. By looking at some examples of premise and hypothesis pairs, we notice that for this dataset, the hypothesis is a summary of the premise, with respect to the quantities, while the premise is longer as it presents more quantities. We can deduce that for the AWPNLI, there will always be a calculation needed between the quantities in the premise, before a comparison can be made to infer the label.

Let's check for duplicates in the dataset, at a premise-hypothesis pair level. Do these duplicates have the same label? If not, which is the pair with the correct label?

In [19]:
```
awp[awp.duplicated(subset=["premise", "hypothesis"])].shape[0]
```

Out[19]:
```
0
```

In [20]:
```
# Turn the premise and hypothesis to lowercase, to ensure we do a case-insensitive check for duplicates as well
awp["premise_lower"] = awp["premise"].str.lower()
awp["hypothesis_lower"] = awp["hypothesis"].str.lower()

awp[awp.duplicated(subset=["premise_lower", "hypothesis_lower"])].shape[0]
```

Out[20]:
```
0
```

Let's inspect the frequency of quantities in the dataset premises and hypotheses

In [21]:
```
awp[["premise_quantities_cnt", "hypothesis_quantities_cnt"]].aggregate(["mean", "std", "min", "max"])
```

Out[21]:

	premise_quantities_cnt	hypothesis_quantities_cnt
mean	2.234072	1.034626
std	0.561652	0.229980
min	1.000000	0.000000
max	4.000000	2.000000

In [22]:
```
awp[awp["hypothesis_quantities_cnt"] == 0][["premise", "hypothesis", "hypothesis_pos"]]
```

Out[22]:

	premise	hypothesis	hypothesis_pos
52	Each of farmer Cunningham 's 6048.0 lambs is e...	5855.0 of Farmer Cunningham 's lambs are black	[[NN, IN, NN, NN, POS, NNS, VBP, JJ]]
53	Each of farmer Cunningham 's 6048.0 lambs is e...	5854.0 of Farmer Cunningham 's lambs are black	[[NN, IN, NN, NN, POS, NNS, VBP, JJ]]
54	A treasure hunter discovered a buried treasure...	5110.0 of the gems were rubies	[[NN, IN, DT, NNS, VBD, NNS]]
55	A treasure hunter discovered a buried treasure...	5108.0 of the gems were rubies	[[NN, IN, DT, NNS, VBD, NNS]]
188	Randy has 78.0 blocks and he uses 19.0 blocks ...	59.0 blocks are left	[[NN, NNS, VBP, VBN]]
532	There was 698.0 children taking a test and 105...	593.0 children had to sit it again	[[JJ, NNS, VBD, TO, VB, PRP, RB]]
533	There was 698.0 children taking a test and 105...	591.0 children had to sit it again	[[JJ, NNS, VBD, TO, VB, PRP, RB]]

Notice that sometimes the role of words is not extracted properly. These sentences seem to have 1 quantity, so the hypotheses have between 1 and 2 quantities, while the premises have between 1 and 4 quantities.

Let's check if the dataset contains features which are not in all datasets

In [23]:
```
awp.columns
```

Out[23]:
```
Index(['sentence2_tokens', 'sentence1_dep_parse', 'sentence1_binary_parse',
       'sentence2_binary_parse', 'sentence1_syntax_parse', 'premise',
       'hypothesis', 'sentence2_syntax_parse', 'hypothesis_pos',
       'sentence2_dep_parse', 'label', 'premise_pos', 'sentence1_tokens',
       'sample_index', 'premise_word_cnt', 'hypothesis_word_cnt',
       'premise_char_cnt', 'hypothesis_char_cnt', 'premise_quantities_cnt',
       'hypothesis_quantities_cnt', 'premise_lower', 'hypothesis_lower'],
      dtype='object')
```

There are no extra features in this dataset to analyze.

NewsNLI dataset

In [24]:
```
news.head()
```

Out[24]:

	sentence2_tokens	annotator_labels	sentence1_tokens	sentence1_dep_parse	sentence2_syntax_parse	Phenomena	sentence1_b
0	[[joey, lepore, says, he, took, photos, of, on...	[entailment, entailment, entailment, neutral, ...	[[lepore, said, he, was, moved, to, photograph...	[[{'dep': 'ROOT', 'dependent': 2, 'governorGlo...	[(ROOT\n (S\n (NP (NN joey) (NN lepore))\n...	[]	(((Lepore)
1	[[darren, sharper, has, been, charged, in, two...	[neutral, entailment, entailment, entailment, ...	[[sharper, ,, 38, ,, faces, rape, charges, in,...	[[{'dep': 'ROOT', 'dependent': 5, 'governorGlo...	[(ROOT\n (S\n (NP\n (NP (NN darren))\...	[Implicit quantity, Arithmetic]	(((((Sharp
2	[[weldon, says, she, 's, a, single, mom, of, t...	[neutral, neutral, entailment, entailment, ent...	[[i, am, the, single, mother, of, three, sons,...	[[{'dep': 'ROOT', 'dependent': 5, 'governorGlo...	[(ROOT\n (S\n (NP (NN weldon))\n (VP (V...	[]	(((((I)) ((a
3	[[the, crash, took, the, lives, of, 79, people...	[neutral, entailment, entailment, entailment, ...	[[in, addition, to, 79, fatalities, ,, some, 1...	[[{'dep': 'ROOT', 'dependent': 11, 'governorGl...	[(ROOT\n (S\n (NP (DT the) (NN crash))\n ...	[]	(((In) (((a
4	[[rip, currents, kill, four, in, alabama, ,, c...	[neutral, entailment, neutral, entailment, ent...	[[treacherous, currents, took, at, least, four...	[[{'dep': 'ROOT', 'dependent': 3, 'governorGlo...	[(ROOT\n (S\n (NP (NN rip) (NNS currents))...	[]	(((Tre currents)

5 rows × 21 columns

In [25]:
```
news['label'].value_counts(normalize=True)
```

Out[25]:
```
label
entailment    0.507231
neutral       0.492769
Name: proportion, dtype: float64
```

We observe a balanced split between the 2 labels of this dataset.

In [26]:
```
sentence_insights(news)

news[["premise", "hypothesis", "label", "premise_word_cnt", "hypothesis_word_cnt", "premise_char_cnt", "hypothes
```

Out[26]:

	premise	hypothesis	label	premise_word_cnt	hypothesis_word_cnt	premise_char_cnt	hypothesis_char_cnt
0	Lepore said he was moved to photograph the slu...	Joey Lepore says he took photos of one guard s...	entailment	20	14	123	73
1	Sharper , 38 , faces rape charges in Arizona a...	Darren Sharper has been charged in two states ...	entailment	22	13	114	76
2	I am the single mother of three sons -- grown ...	Weldon says she 's a single mom of three flour...	entailment	17	16	78	85
3	In addition to 79 fatalities , some 170 passen...	The crash took the lives of 79 people and inju...	entailment	12	12	65	58
4	Treacherous currents took at least four lives ...	Rip currents kill four in Alabama , close beac...	entailment	23	11	128	60

In [27]:
```
news[["premise_word_cnt", "hypothesis_word_cnt", "premise_char_cnt", "hypothesis_char_cnt"]].aggregate(["mean",
```

Out[27]:

	premise_word_cnt	hypothesis_word_cnt	premise_char_cnt	hypothesis_char_cnt
mean	22.330579	11.984504	120.600207	66.581612
std	8.180754	2.999615	46.449696	15.740119

We observe that the premises are on average twice as long as the hypothesis, with respect to the number of words and characters. By inspecting some of the premise-hypothesis pairs, we notice that the hypothesis is usually a shorter, rephrased version of the premise (similar to a summary of a long sentence). However, in contrast to the AWPNLI dataset, there are not necessarily calculations that need to be done in either of the 2 sentences.

Let's check for duplicates in the dataset, at a premise-hypothesis pair level. Do these duplicates have the same label? If not, which is the pair with the correct label?

In [28]:
```
news[news.duplicated(subset=["premise", "hypothesis"])].shape[0]
```

Out[28]: 5

In [29]:
```
news[news.duplicated(subset=["premise", "hypothesis", "label"])].shape[0]
```

Out[29]: 5

It seems like the duplicates have the same label

In [30]:
```
# Turn the premise and hypothesis to lowercase, to ensure we do a case-insensitive check for duplicates as well
news["premise_lower"] = news["premise"].str.lower()
news["hypothesis_lower"] = news["hypothesis"].str.lower()

news[news.duplicated(subset=["premise_lower", "hypothesis_lower"])].shape[0]
```

Out[30]: 5

In [31]:
```
news[news.duplicated(subset=["premise", "hypothesis"])][["premise", "hypothesis", "label"]]
```

Out[31]:

	premise	hypothesis	label
41	Mycoskie had already started four other busine...	Blake Mycoskie had launched four other start-u...	entailment
67	Cobb declined two requests from CNN to respond...	Cobb declined two requests to speak with CNN f...	entailment
151	In fact , Wernick had only seen one zombie fil...	One of film 's writers had seen just one zombi...	entailment
300	There were no reports of serious injuries , bu...	At least 8 reported arrested , but no reports ...	entailment
416	42 percent of homeless children are younger th...	Study says 42 percent of homeless children are...	entailment

Let's inspect the frequencies of quantities in the premises and hypotheses

In [32]:
```
news[["premise_quantities_cnt", "hypothesis_quantities_cnt"]].aggregate(["mean", "std", "min", "max"])
```

Out[32]:

	premise_quantities_cnt	hypothesis_quantities_cnt
mean	1.622934	1.380165
std	1.033748	0.736196
min	0.000000	0.000000
max	12.000000	8.000000

In [33]:
```
news_no_quantities = news[(news["premise_quantities_cnt"] == 0) | (news["hypothesis_quantities_cnt"] == 0)][["p
news_no_quantities
```

Out[33]:

	premise	hypothesis	sample_index
94	Shaffer : Just to be clear , I was offered the...	Shaffer was offered chance to play Jerry Seinf...	94
127	That inmate and the county worker were undergo...	Two of the injured were undergoing emergency s...	127
178	But terrarium gardens and other tiny plant pro...	Terrariums and other small plant projects are ...	178
317	Jiang has became a celebrity , followed by loc...	Newspaper headline hails her as " China 's Mo...	317
364	" Jeremy Lin is a marketing dream come true ,...	Lin is a " marketing dream come true , " one...	364
507	(CNN) -- Tony Gwynn , a Hall of Fame outfiel...	Gwynn died at 54 after a long battle with sali...	507
513	After they complete their sentence , the pair ...	The two Britons will be deported after they co...	513
530	In the latest attack , a parked motorcycle bom...	Motorcycle bomb kills six in Sunni neighborhoo...	530
569	(CNET.com) -- The HP Pavilion Media Center T...	The HP Pavilion Media Center TV m8120n retails...	569
592	The latest trend is theaters offering " luxur...	Premium screening rooms offer cocktails , wine...	592
671	(CNN) -- Indonesian police are searching for...	More than 200 inmates escaped from Indonesian ...	671
697	While the cardinal-electors are locked in the ...	115 cardinal-electors are gathered in the Sist...	697
778	Hundreds of thousands took to the streets in B...	At the height of the war , 46,000 British troo...	778
802	The contest rules spelled out that NASA reserv...	NASA reserves right to pick name for Node 3	802
810	Tamer was at the same demonstration Hamza atte...	Tamer Mohammed al Sharey , 15 , disappeared at...	810
875	Bobby Jindal declared a statewide state of eme...	Florida governor declares a state of emergency...	875
886	The incident started after South Korean comman...	South Korea has seized the ship and its nine s...	886
887	The second explosion took place at a crowded b...	The earlier explosion injured 12 people at a N...	887
894	The British island group of Tristan da Cunha s...	The islands of Tristan da Cunha sit 1,750 mile...	894
900	Five service members hurt , building damaged i...	He said the attack included rockets , small ar...	900
904	With the Eastern Cape being so key to the coun...	The Eastern Cape provides 51 % of South Africa...	904
909	Stone has come up with a name for the new stat...	He has drawn plans for 13 counties to form the...	909
911	With a win in Dubai , Stenson would become the...	Henrik Stenson shoots an eight-under-par 64 to...	911
919	I 'd rather be na ve , heartfelt and hopeful t...	" Call me na ve , " Vedder said in website post	919
951	Kapoor and Eroshevich were each also charged w...	All three charged with giving " a controlled ...	951
960	Armed officers were confronted by a pack of do...	Four dogs were shot by armed police officers a...	960
967	Sponseller graduated from The Citadel and is a...	Tom Sponseller , 61 , is head of the South Car...	967

In [34]:
```
news_no_quantities[["premise", "hypothesis", "sample_index"]].to_excel("NewsNLI_no_quantities.xlsx")
```

Manual inspection of these samples reveals that indeed there are no quantities in them, either in numerical or verbal format. However, if at least one of the premise or hypothesis does contain a quantity, the pair should not be discarded. These could represent "neutral" samples, where the label is not necessarily inferred on a quantitative basis, but on the lack of details in one sentence (usually the premise) to support the quantitative details in the other (usually the hypothesis).

In [35]:
```
pairs_no_quantities = news[(news["premise_quantities_cnt"] == 0) & (news["hypothesis_quantities_cnt"] == 0)][["
pairs_no_quantities
```

Out[35]:

	premise	hypothesis	sample_index
178	But terrarium gardens and other tiny plant pro...	Terrariums and other small plant projects are ...	178
919	I 'd rather be na ve , heartfelt and hopeful t...	" Call me na ve , " Vedder said in website post	919

These 2 samples could be dropped as they do not contain any quantitative information so they are not part of the type of sentences QNLI focuses on.

Let's inspect the data on the annotator labels - for how many samples were there disagreements between annotators?

In [36]:
```
news["annotator_unique_labels"].value_counts(normalize=True)
```

Out[36]:
```
annotator_unique_labels
2    0.646694
1    0.351240
3    0.002066
Name: proportion, dtype: float64
```

It seems like in almost 65% of cases, there was a disagreement between the annotators. This can also indicate a higher complexity of the sentences in this dataset.

In [37]:
```
news.groupby("annotator_unique_labels")["label"].value_counts(normalize=True)
```

Out[37]:
```
annotator_unique_labels  label
1                        neutral       0.523529
                         entailment    0.476471
2                        entailment    0.523962
                         neutral       0.476038
3                        entailment    0.500000
                         neutral       0.500000
Name: proportion, dtype: float64
```

It also looks like the disagreements were almost equally split between samples from both categories.

Let's check if the dataset contains features which are not in all datasets

In [38]:
```
news.columns
```

Out[38]:
```
Index(['sentence2_tokens', 'annotator_labels', 'sentence1_tokens',
       'sentence1_dep_parse', 'sentence2_syntax_parse', 'Phenomena',
       'sentence1_binary_parse', 'Hard', 'sentence2_parse',
       'sentence2_binary_parse', 'sentence1_syntax_parse', 'premise',
       'hypothesis', 'sentence1_parse', 'genre', 'hypothesis_pos',
       'sentence2_dep_parse', 'label', 'premise_pos', 'PairID', 'sample_index',
       'premise_word_cnt', 'hypothesis_word_cnt', 'premise_char_cnt',
       'hypothesis_char_cnt', 'premise_quantities_cnt',
       'hypothesis_quantities_cnt', 'annotator_unique_labels', 'premise_lower',
       'hypothesis_lower'],
      dtype='object')
```

In [39]:
```
news["Phenomena"] = news["Phenomena"].fillna("[]")
news["Hard"] = news["Hard"].fillna("Unknown")
```

In [40]:
```
extra_columns = ["Phenomena", "Hard", "genre"]
for column in extra_columns:
    print(f"###############")
    print(news[column].value_counts())
```

```
###############
Phenomena
[]                                                461
[]                                                403
[Numeration]                                       32
[QC]                                               11
[Count]                                             8
[Implicit quantity,  Arithmetic]                    6
[Quantifiers]                                       5
[Unit conversion]                                   4
[Numeration,  Arithmetic]                           2
[Arithmetic]                                        2
[Named Set]                                         2
[Ratios]                                            2
[Named set resolution]                              2
[SETS]                                              2
[Numeration,  Unit conversion]                      2
[Ordinality]                                        2
[Quantity conversion]                               2
[Quantity conversion,  Quantifiers]                 2
[Quantifiers,  Ranges]                              1
[MULIPLE]                                           1
[COUNT]                                             1
[Sets]                                              1
[Reasoning]                                         1
[Quantifiers,  Numeration]                          1
[Named set resolution,  Implicit quantity]          1
[Quantifiers,  Quantity conversion]                 1
[Numeration,  Quantity conversion]                  1
[Approximation]                                     1
[Implicit quantity,  Numeration]                    1
[Multi-hop reasoning,  Quantity conversion]         1
[Arithmetic,  Numeration]                           1
[Named set resolution,  Arithmetic]                 1
[Quantity conversion,  Approximation]               1
[Implicit quantity,  Arithmetic,  Numeration]       1
[Approximation,  Quantifiers]                       1
[Quantifier,  Geography]                            1
Name: count, dtype: int64
###############
Hard
Yes    507
No     461
Name: count, dtype: int64
###############
genre
News    968
Name: count, dtype: int64
```

It seems like the 'genre' column is not informative, it points to the source of the samples, namely news articles.

The 'Hard' column is of Boolean nature, it may indicate if a certain example is harder to classify (more complex), but this is only an assumption. There is no information in the original paper about this column or its meaning.

The 'Phenomena' column seems to assign categories to some of the samples, of quantitative phenomena. It would be interesting to analyze the results on the samples with Phenomena vs the samples without, to see if there is any discrepancy. The ratio of samples with phenomena is relatively low.

In [41]:
```
## are 'Hard' examples the ones with 'Phenomena'?

news[news['Hard'] == 'Yes']["Phenomena"].value_counts()
```

Out[41]:
```
Phenomena
[]                                                  403
[Numeration]                                         32
[QC]                                                 11
[Count]                                               8
[Implicit quantity,  Arithmetic]                      6
[Quantifiers]                                         5
[Unit conversion]                                     4
[SETS]                                                2
[Ratios]                                              2
[Named Set]                                           2
[Numeration,  Arithmetic]                             2
[Named set resolution]                                2
[Arithmetic]                                          2
[Ordinality]                                          2
[Numeration,  Unit conversion]                        2
[Quantity conversion,  Quantifiers]                   2
[Quantity conversion]                                 2
[Arithmetic,  Numeration]                             1
[Implicit quantity,  Arithmetic,  Numeration]         1
[Sets]                                                1
[COUNT]                                               1
[MULIPLE]                                             1
[Reasoning]                                           1
[Approximation,  Quantifiers]                         1
[Quantifiers,  Ranges]                                1
[Named set resolution,  Arithmetic]                   1
[Named set resolution,  Implicit quantity]            1
[Quantifiers,  Quantity conversion]                   1
[Numeration,  Quantity conversion]                    1
[Approximation]                                       1
[Quantifiers,  Numeration]                            1
[Quantity conversion,  Approximation]                 1
[Implicit quantity,  Numeration]                      1
[Multi-hop reasoning,  Quantity conversion]           1
[Quantifier,  Geography]                              1
Name: count, dtype: int64
```

In [42]:
```
news[news['Hard'] == 'No']["Phenomena"].value_counts()
```

Out[42]:
```
Phenomena
[]    461
Name: count, dtype: int64
```

All samples with phenomena are categorized as 'hard', but there are also samples with no-phenomena out in the same category. It remains unclear what the 'Hard' column could represent.

Reddit dataset

In [43]:
```
reddit.head()
```

Out[43]:

	sentence2_tokens	sentence1_tokens	sentence1_dep_parse	sentence2_syntax_parse	sentence1_binary_parse	sentence2_parse
0	[[sensex, and, nifty, up, ,, 2, sept, nifty, s...	[[stocks, nifty, future, call, today, :, sense...	[[{'dep': 'ROOT', 'dependent': 4, 'governorGlo...	[(ROOT\n (NP\n (NP\n (NP (NN sensex))...	((NP-TMP ((stocks)) ((nifty) (future ...	(NP (NP (NP (NP (NNP Sensex) (CC and) (NNP Nif...
1	[[at, davos, ,, wall, street, billionaire, ste...	[[at, davos, ,, financial, billionaire, schwar...	[[{'dep': 'ROOT', 'dependent': 19, 'governorGl...	[(ROOT\n (S\n (PP (IN at)\n (NP (NNP ...	(((At) ((DAVOS))) (, ,) (((Financ...	(S (PP (IN At) (NP (NNP Davos))) (, ,) (NP (NP...
2	[[sensex, down, 74.58, points, ,, nifty, futur...	[[sensex, nifty, up, ,, today, stocks, nifty, ...	[[{'dep': 'ROOT', 'dependent': 2, 'governorGlo...	[(ROOT\n (S\n (NP\n (NP (NN sensex))\...	(((((SENSEX) (Nifty)) ((up))) (,...	(SINV (VP (VB Sensex) (PRT (RP down)) (NP (NP ...
3	[[at, davos, ,, wall, street, billionaire, mr,...	[[at, davos, ,, financial, billionaire, schwar...	[[{'dep': 'ROOT', 'dependent': 19, 'governorGl...	[(ROOT\n (S\n (PP (IN at)\n (NP (NNP ...	(((At) ((DAVOS))) (, ,) (((Financ...	(S (PP (IN At) (NP (NNP Davos))) (, ,) (NP (NP...
4	[[stocks, nifty, future, call, today, :, sense...	[[sensex, and, nifty, up, ,, 2, sept, nifty, s...	[[{'dep': 'ROOT', 'dependent': 1, 'governorGlo...	[(ROOT\n (FRAG\n (NP-TMP\n (NP (NNS s...	(((((Sensex) (and) (Nifty)) ((up ...	(FRAG (NP-TMP (NP (NNS stocks)) (NP (JJ nifty)...

In [44]:
```
reddit['label'].value_counts(normalize=True)
```

Out[44]:
```
label
entailment       0.584
neutral          0.340
contradiction    0.076
Name: proportion, dtype: float64
```

We notice an imbalance in this dataset between the 3 labels, with the contradiction label representing less than 10% of the samples (specifically 7.6%). The entailment label is represents the majority, followed by neutral.

In [45]:
```
sentence_insights(reddit)

reddit[["premise", "hypothesis", "label", "premise_word_cnt", "hypothesis_word_cnt", "premise_char_cnt", "hypot
```

```
There is no data on the annotator labels for this dataset.
```

Out[45]:

	premise	hypothesis	label	premise_word_cnt	hypothesis_word_cnt	premise_char_cnt	hypothesis_char_cnt
0	stocks nifty future call today: Sensex Weak an...	Sensex and Nifty up, 2 sept Nifty stock market...	contradiction	28	30	153	167
1	At DAVOS, Financial Billionaire Schwartzman, w...	At Davos, Wall Street Billionaire Steven Schwa...	neutral	22	24	146	160
2	SENSEX Nifty up, Today stocks nifty future tra...	Sensex down 74.58 points, Nifty future tips, T...	contradiction	27	26	153	158
3	At DAVOS, Financial Billionaire Schwartzman, w...	At Davos, Wall Street Billionaire Mr Schwartzf...	entailment	22	24	146	154
4	Sensex and Nifty up, 2 sept Nifty stock market...	stocks nifty future call today: Sensex Weak an...	contradiction	30	28	167	153

In [46]:
```
reddit[["premise_word_cnt", "hypothesis_word_cnt", "premise_char_cnt", "hypothesis_char_cnt"]].aggregate(["mean
```

Out[46]:

	premise_word_cnt	hypothesis_word_cnt	premise_char_cnt	hypothesis_char_cnt
mean	11.960000	11.460000	69.448000	65.660000
std	4.833617	5.144537	30.102757	31.356488

We notice that the frequency of words and characters is very similar between the premises and hypotheses.

Let's check for duplicates in the dataset, at a premise-hypothesis pair level. Do these duplicates have the same label? If not, which is the pair with the correct label?

In [47]:
```
reddit[reddit.duplicated(subset=["premise", "hypothesis"])].shape[0]
```

Out[47]: 3

In [48]:
```
# do the duplicates have the same label?
reddit[reddit.duplicated(subset=["premise", "hypothesis", "label"])].shape[0]
```

Out[48]: 3

In [49]:
```
# Turn the premise and hypothesis to lowercase, to ensure we do a case-insensitive check for duplicates as well
reddit["premise_lower"] = reddit["premise"].str.lower()
reddit["hypothesis_lower"] = reddit["hypothesis"].str.lower()

reddit[reddit.duplicated(subset=["premise_lower", "hypothesis_lower"])].shape[0]
```

Out[49]: 3

In [50]:
```
reddit[reddit.duplicated(subset=["premise", "hypothesis"])][["premise", "hypothesis", "sample_index"]]
```

Out[50]:

	premise	hypothesis	sample_index
54	U.S. economy added 161,000 jobs in October as ...	U.S. Economy Grew by 161,000 Jobs in October; ...	54
125	Wages Salaries jump by 3.1 percent; highest in...	Wages and salaries jump by 3.1%, highest level...	125
127	U.S. economy off to slow start in 2017 under T...	G.D.P. Report Shows U.S. Economy Off to Slow S...	127

These 3 samples should be discarded from the training/testing datasets.

Let's inspect the frequency of quantities in the dataset premises and hypotheses

In [51]:
```
reddit[["premise_quantities_cnt", "hypothesis_quantities_cnt"]].aggregate(["mean", "std", "min", "max"])
```

Out[51]:

	premise_quantities_cnt	hypothesis_quantities_cnt
mean	1.632000	1.552000
std	0.811919	0.811127
min	0.000000	0.000000
max	4.000000	6.000000

In [52]:
```
reddit_no_quantities = reddit[(reddit["premise_quantities_cnt"] == 0) | (reddit["hypothesis_quantities_cnt"] ==
reddit_no_quantities
```

Out[52]:

	premise	hypothesis	sample_index
33	Based off 1st time unemployment claims, the Ju...	Based off of 1st unemployment reports the jobs...	33
34	Based off of 1st unemployment reports the jobs...	Based off 1st time unemployment claims, the Ju...	34
163	Dow Closes Above 18K for First Time Since July	Dow closes above 18000 for first time in 9 months	163
182	Dow closes above 18000 for first time in 9 months	Dow Closes Above 18K for First Time Since July	182
208	Home ownership falls to lowest level since the...	Home ownership hits lowest level since 1965	208

Manual inspection of these samples indicates that they actually contain quantities, so they should not be discarded from the dataset.

Let's check if the dataset contains features which are not in all datasets

In [53]:
```
reddit.columns
```

Out[53]:
```
Index(['sentence2_tokens', 'sentence1_tokens', 'sentence1_dep_parse',
       'sentence2_syntax_parse', 'sentence1_binary_parse', 'sentence2_parse',
       'sentence2_binary_parse', 'sentence1_syntax_parse', 'premise',
       'hypothesis', 'sentence1_parse', 'genre', 'hypothesis_pos',
       'sentence2_dep_parse', 'label', 'premise_pos', 'PairID', 'sample_index',
       'premise_word_cnt', 'hypothesis_word_cnt', 'premise_char_cnt',
       'hypothesis_char_cnt', 'premise_quantities_cnt',
       'hypothesis_quantities_cnt', 'premise_lower', 'hypothesis_lower'],
      dtype='object')
```

In [54]:
```
# only the 'genre' column is an extra column, let's inspect its values
reddit['genre'].value_counts()
```

Out[54]:
```
genre
Economic News    250
Name: count, dtype: int64
```

RTE dataset

In [55]:
```
rte.head()
```

Out[55]:

	sentence2_tokens	annotator_labels	sentence1_dep_parse	sentence1_binary_parse	sentence2_binary_parse	sentence1_syntax_p
0	[[accardo, composed, 24, caprices, .]]	[neutral, neutral, neutral, neutral, neutral]	[[{'dep': 'ROOT', 'dependent': 4, 'governorGlo...	In 1956 Accardo won the Geneva Competition and...	Accardo composed 24 Caprices .	[(ROOT\n (S\n (PP in)\n (NP (CD
1	[[golinkin, has, written, eighteen, books, .]]	[neutral, neutral, neutral, neutral, neutral]	[[{'dep': 'ROOT', 'dependent': 5, 'governorGlo...	David Golinkin is the editor or author of eigh...	Golinkin has written eighteen books .	[(ROOT\n (S\n (NP david) (NN golink
2	[[david, golinkin, is, the, author, of, dozen,...	[neutral, neutral, neutral, neutral, neutral]	[[{'dep': 'ROOT', 'dependent': 5, 'governorGlo...	David Golinkin is single-handedly responsible ...	David Golinkin is the author of dozen of respo...	[(ROOT\n (S\n (NP david) (NN golink
3	[[reinsdorf, was, the, chairman, of, the, whit...	[entailment, entailment, entailment, entailmen...	[[{'dep': 'ROOT', 'dependent': 16, 'governorGl...	During Reinsdorf 's 24 seasons as chairman of ...	Reinsdorf was the chairman of the White Sox fo...	[(ROOT\n (S\n (PP during)\n (NP
4	[[the, white, sox, have, won, 24, championship...	[neutral, neutral, neutral, neutral, neutral]	[[{'dep': 'ROOT', 'dependent': 16, 'governorGl...	During Reinsdorf 's 24 seasons as chairman of ...	The White Sox have won 24 championships .	[(ROOT\n (S\n (PP during)\n (NP

In [56]:
```
rte['label'].value_counts(normalize=True)
```

Out[56]:
```
label
neutral       0.578313
entailment    0.421687
Name: proportion, dtype: float64
```

The RTE_Quant dataset is relatively balanced between the 2 labels

In [57]:
```
sentence_insights(rte)

rte[["premise", "hypothesis", "label", "premise_word_cnt", "hypothesis_word_cnt", "premise_char_cnt", "hypothes
```

Out[57]:

	premise	hypothesis	label	premise_word_cnt	hypothesis_word_cnt	premise_char_cnt	hypothesis_char_cnt
0	In 1956 Accardo won the Geneva Competition and...	Accardo composed 24 Caprices .	neutral	52	5	281	30
1	David Golinkin is the editor or author of eigh...	Golinkin has written eighteen books .	neutral	22	6	113	37
2	David Golinkin is single-handedly responsible ...	David Golinkin is the author of dozen of respo...	neutral	37	22	239	123
3	During Reinsdorf 's 24 seasons as chairman of ...	Reinsdorf was the chairman of the White Sox fo...	entailment	31	12	176	60
4	During Reinsdorf 's 24 seasons as chairman of ...	The White Sox have won 24 championships .	neutral	31	8	176	41

In [58]:
```
rte[["premise_word_cnt", "hypothesis_word_cnt", "premise_char_cnt", "hypothesis_char_cnt"]].aggregate(["mean",
```

Out[58]:

	premise_word_cnt	hypothesis_word_cnt	premise_char_cnt	hypothesis_char_cnt
mean	32.138554	11.590361	176.753012	61.246988
std	12.420856	4.819245	69.784728	26.422542

We notice that the premises are usually much larger than the hypotheses, around 3 times larger, respectively. Similarly to the NewsNLI dataset, the hypothesis seems to be a shorter, rephrased version of the premise, so a kind of summary.

Let's check for duplicates in the dataset, at a premise-hypothesis pair level. Do these duplicates have the same label? If not, which is the pair with the correct label?

In [59]:
```
rte[rte.duplicated(subset=["premise", "hypothesis"])].shape[0]
```

Out[59]: 1

In [60]:
```
# are the duplicates still duplicates at a label level as well?
rte[rte.duplicated(subset=["premise", "hypothesis", "label"])].shape[0]
```

Out[60]: 1

In [61]:
```
rte[rte.duplicated(subset=["premise", "hypothesis"])][["premise", "hypothesis", "sample_index"]]
```

Out[61]:

	premise	hypothesis	sample_index
110	Phil Mickelson finished a triumphant week in h...	Mickelson won by five shots last week , the la...	110

This sample should be discarded from the training/testing sets.

Let's inspect the frequency of quantities in the dataset premises and hypotheses

In [62]:
```
rte[["premise_quantities_cnt", "hypothesis_quantities_cnt"]].aggregate(["mean", "std", "min", "max"])
```

Out[62]:

	premise_quantities_cnt	hypothesis_quantities_cnt
mean	1.987952	1.253012
std	1.190813	0.727692
min	0.000000	0.000000
max	6.000000	5.000000

In [63]:
```
rte_no_quantities = rte[(rte["premise_quantities_cnt"] == 0) | (rte["hypothesis_quantities_cnt"] == 0)][["premi
rte_no_quantities
```

Out[63]:

	premise	hypothesis	sample_index
2	David Golinkin is single-handedly responsible ...	David Golinkin is the author of dozen of respo...	2
6	Dr. Felix Soto Toro (born 1967 in Guaynabo , ...	Soto Toro invented a 3D measuring system .	6
21	The 8,568-meter Mt . Kanchenjunga , the third ...	Kanchenjunga is 8586 meters high .	21
38	A federal judge sentenced an apparently stunne...	Milken was given a 10-year sentence .	38
46	Prosecutions tended to be more aggressive and ...	Bilking a large number of people out of millio...	46
47	Even though there is some evidence that suppor...	It is predicted that as of 1994 , a referendum...	47
49	A Los Angeles federal court judge Monday impos...	A Los Angeles federal judge imposed a 15-year ...	49
69	Due to these effects , a person who has consum...	Half of road-traffic deaths are caused by alco...	69
76	Monday , when the hearings begin , the Palesti...	Israelis will demonstrate and a counter-demons...	76
88	Israeli security forces seized large amounts o...	The forces took millions of shekels in cash fr...	88
92	Two car bombs explode near a police station ou...	A pair of car bombs explode near government of...	92
107	Commandos stormed a school Friday in southern ...	The total number of hostages held in the schoo...	107
127	GUS on Friday disposed of its remaining home s...	Wehkamp cost % u20AC390m .	127
129	Last week , saw the fall of the Dutch right wi...	Three parties form a Dutch coalition government .	129
133	Of all the national park lands in the United S...	The Everglades is 50-mile wide .	133
138	It is outstripped only by Denmark , the Nether...	12 members of the European Union use the Euro ...	138
141	A seven-member Tibetan mountaineering team con...	Kanchenjunga is 8586 meters high .	141
142	The 10-men team is expected to arrive at the f...	Kanchenjunga is 8586 meters high .	142
149	Police in Rio de Janeiro arrested five men and...	Millions of dollars of art were recovered , in...	149
150	Stolen Warhol works recovered : Amsterdam poli...	Millions of dollars of art were recovered , in...	150
156	More than 6,400 migratory birds and other anim...	Animals have died by the thousands from drinki...	156

In [64]:
```
# let's inspect these samples manually to decide if any should be discarded
rte_no_quantities.to_excel("RTE_Quant_no_quantities.xlsx")
```

Let's inspect the data on the annotator labels - for how many samples were there disagreements between annotators?

In [65]:
```
rte["annotator_unique_labels"].value_counts()
```

Out[65]:
```
annotator_unique_labels
1    166
Name: count, dtype: int64
```

It appears like annotators were never in disagreement over the label of a sample.

Let's check if the dataset contains features which are not in all datasets

In [66]:
```
rte.columns
```

Out[66]:
```
Index(['sentence2_tokens', 'annotator_labels', 'sentence1_dep_parse',
       'sentence1_binary_parse', 'sentence2_binary_parse',
       'sentence1_syntax_parse', 'premise', 'hypothesis',
       'sentence2_syntax_parse', 'genre', 'hypothesis_pos',
       'sentence2_dep_parse', 'label', 'premise_pos', 'sentence1_tokens',
       'sample_index', 'premise_word_cnt', 'hypothesis_word_cnt',
       'premise_char_cnt', 'hypothesis_char_cnt', 'premise_quantities_cnt',
       'hypothesis_quantities_cnt', 'annotator_unique_labels'],
      dtype='object')
```

In [67]:
```
rte['genre'].value_counts()
```

Out[67]:
```
genre
news    166
Name: count, dtype: int64
```

Besides 'genre', there are no extra columns in this dataset to analyze.

StressTest dataset

In [68]:
```
stress.head()
```

Out[68]:

	sentence2_tokens	sentence1_dep_parse	sentence1_binary_parse	sentence2_binary_parse	sentence1_syntax_parse	premise	h
0	[[if, joe, goes, with, her, more, than, 1, yea...	[[{'dep': 'ROOT', 'dependent': 3, 'governorGlo...	If Joe goes with her 6 years old twin brothers...	If Joe goes with her more than 1 years old twi...	[(ROOT\n (SBAR (IN if)\n (S\n (S\n ...	If Joe goes with her 6 years old twin brothers...	I
1	[[if, joe, goes, with, her, 6, years, old, twi...	[[{'dep': 'ROOT', 'dependent': 3, 'governorGlo...	If Joe goes with her more than 1 years old twi...	If Joe goes with her 6 years old twin brothers...	[(ROOT\n (SBAR (IN if)\n (S\n (S\n ...	If Joe goes with her more than 1 years old twi...	I
2	[[if, joe, goes, with, her, less, than, 6, yea...	[[{'dep': 'ROOT', 'dependent': 3, 'governorGlo...	If Joe goes with her 6 years old twin brothers...	If Joe goes with her less than 6 years old twi...	[(ROOT\n (SBAR (IN if)\n (S\n (S\n ...	If Joe goes with her 6 years old twin brothers...	I
3	[[tim, has, less, than, 750, pounds, of, cemen...	[[{'dep': 'ROOT', 'dependent': 2, 'governorGlo...	Tim has 350 pounds of cement in 100 , 50 , and...	Tim has less than 750 pounds of cement in 100 ...	[(ROOT\n (S\n (NP (NN tim))\n (VP (VBZ ...	Tim has 350 pounds of cement in 100 , 50 , and...	7
4	[[tim, has, 350, pounds, of, cement, in, 100, ...	[[{'dep': 'ROOT', 'dependent': 2, 'governorGlo...	Tim has less than 750 pounds of cement in 100 ...	Tim has 350 pounds of cement in 100 , 50 , and...	[(ROOT\n (S\n (NP (NN tim))\n (VP (VBZ ...	Tim has less than 750 pounds of cement in 100 ...	3 ir

In [69]:
```
stress['label'].value_counts(normalize=True)
```

Out[69]:
```
label
entailment       0.333333
neutral          0.333333
contradiction    0.333333
Name: proportion, dtype: float64
```

StressTest is balanced with respect to the distribution of labels.

In [70]:
```
sentence_insights(stress)
```

```
stress[["premise", "hypothesis", "label", "premise_word_cnt", "hypothesis_word_cnt", "premise_char_cnt", "hypotl
```

There is no data on the annotator labels for this dataset.

Out[70]:

	premise	hypothesis	label	premise_word_cnt	hypothesis_word_cnt	premise_char_cnt	hypothesis_char_cnt
0	If Joe goes with her 6 years old twin brothers...	If Joe goes with her more than 1 years old twi...	entailment	19	21	84	94
1	If Joe goes with her more than 1 years old twi...	If Joe goes with her 6 years old twin brothers...	neutral	21	19	94	84
2	If Joe goes with her 6 years old twin brothers...	If Joe goes with her less than 6 years old twi...	contradiction	19	21	84	94
3	Tim has 350 pounds of cement in 100 , 50 , and...	Tim has less than 750 pounds of cement in 100 ...	entailment	15	17	60	70
4	Tim has less than 750 pounds of cement in 100 ...	Tim has 350 pounds of cement in 100 , 50 , and...	neutral	17	15	70	60

In [71]:
```
stress[["premise_word_cnt", "hypothesis_word_cnt", "premise_char_cnt", "hypothesis_char_cnt"]].aggregate(["mean
```

Out[71]:

	premise_word_cnt	hypothesis_word_cnt	premise_char_cnt	hypothesis_char_cnt
mean	20.905082	21.247367	95.760269	97.432464
std	9.979773	10.002253	45.704475	45.769032

It looks like the premises and hypotheses are of almost equal length in this dataset. By inspecting some pairs, it appears that the difference between the premises and hypotheses in this dataset is a change of the quantity and/or the addition or removal of a quantifier (i.e. either the premise gives and estimate of a quantity and the hypothesis gives a fixed value or the other way around).

Let's check for duplicates in the dataset, at a premise-hypothesis pair level. Do these duplicates have the same label? If not, which is the pair with the correct label?

In [72]:
```
stress[stress.duplicated(subset=["premise", "hypothesis"])].shape[0]
```

Out[72]: 643

In [73]:
```
# Turn the premise and hypothesis to lowercase, to ensure we do a case-insensitive check for duplicates as well
stress["premise_lower"] = stress["premise"].str.lower()
stress["hypothesis_lower"] = stress["hypothesis"].str.lower()

stress[stress.duplicated(subset=["premise_lower", "hypothesis_lower"])].shape[0]
```

Out[73]: 649

It seems like there is a significant number of duplicates in this dataset, and case-sensitivity must be considered, as it discovers an extra 6 duplicates. let's check if these duplicates have different labels.

In [74]:
```
stress[stress.duplicated(subset=["premise_lower", "hypothesis_lower", "label"])].shape[0]
```

Out[74]: 649

In [75]:
```
# what % of the total samples are duplicates that must be dropped?
649 / stress.shape[0]
```

Out[75]: 0.08543970510795156

The 649 duplicates will be discarded from the final training / testing sets.

Let's inspect the frequency of quantities in the dataset premises and hypotheses

In [76]:
```
stress[["premise_quantities_cnt", "hypothesis_quantities_cnt"]].aggregate(["mean", "std", "min", "max"])
```

Out[76]:

	premise_quantities_cnt	hypothesis_quantities_cnt
mean	2.098736	2.100316
std	1.275481	1.274945
min	0.000000	0.000000
max	9.000000	9.000000

In [77]:
```
stress[(stress["premise_quantities_cnt"] == 0) | (stress["hypothesis_quantities_cnt"] == 0)][["premise", "hypot
```

Out[77]:

	premise	hypothesis	sample_index
1068	James took a 3 - hour bike ride	James took a less than 4 - hour bike ride	1068
1069	James took a less than 4 - hour bike ride	James took a 3 - hour bike ride	1069
1070	James took a 3 - hour bike ride	James took a 1 - hour bike ride	1070
2343	James took a 3 - hour bike ride	James took a less than 8 - hour bike ride	2343
2344	James took a less than 8 - hour bike ride	James took a 3 - hour bike ride	2344
2345	James took a 3 - hour bike ride	James took a more than 3 - hour bike ride	2345
3051	James took a 3 - hour bike ride	James took a more than 1 - hour bike ride	3051
3052	James took a more than 1 - hour bike ride	James took a 3 - hour bike ride	3052
3053	James took a 3 - hour bike ride	James took a more than 3 - hour bike ride	3053
5097	Jack took a 3 - hour bike ride	Jack took a less than 7 - hour bike ride	5097
5098	Jack took a less than 7 - hour bike ride	Jack took a 3 - hour bike ride	5098
5099	Jack took a 3 - hour bike ride	Jack took a 2 - hour bike ride	5099
5154	James took a 3 - hour bike ride	James took a less than 7 - hour bike ride	5154
5155	James took a less than 7 - hour bike ride	James took a 3 - hour bike ride	5155
5156	James took a 3 - hour bike ride	James took a 8 - hour bike ride	5156
5943	James took a 3 - hour bike ride	James took a more than 2 - hour bike ride	5943
5944	James took a more than 2 - hour bike ride	James took a 3 - hour bike ride	5944
5945	James took a 3 - hour bike ride	James took a less than 3 - hour bike ride	5945
7071	James took a 3 - hour bike ride	James took a more than 1 - hour bike ride	7071
7072	James took a more than 1 - hour bike ride	James took a 3 - hour bike ride	7072
7073	James took a 3 - hour bike ride	James took a 8 - hour bike ride	7073

Manual inspection of these samples indicates that they actually contain quantities and thus should not be discarded.

Let's check if the dataset contains features which are not in all datasets

In [78]:
```
stress.columns
```

Out[78]:
```
Index(['sentence2_tokens', 'sentence1_dep_parse', 'sentence1_binary_parse',
       'sentence2_binary_parse', 'sentence1_syntax_parse', 'premise',
       'hypothesis', 'sentence2_syntax_parse', 'hypothesis_pos',
       'sentence2_dep_parse', 'label', 'premise_pos', 'sentence1_tokens',
       'sample_index', 'premise_word_cnt', 'hypothesis_word_cnt',
       'premise_char_cnt', 'hypothesis_char_cnt', 'premise_quantities_cnt',
       'hypothesis_quantities_cnt', 'premise_lower', 'hypothesis_lower'],
      dtype='object')
```

There are no extra columns to be analyzed

Let's generate word clouds of the premises and hypotheses.

In [79]:
```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
```

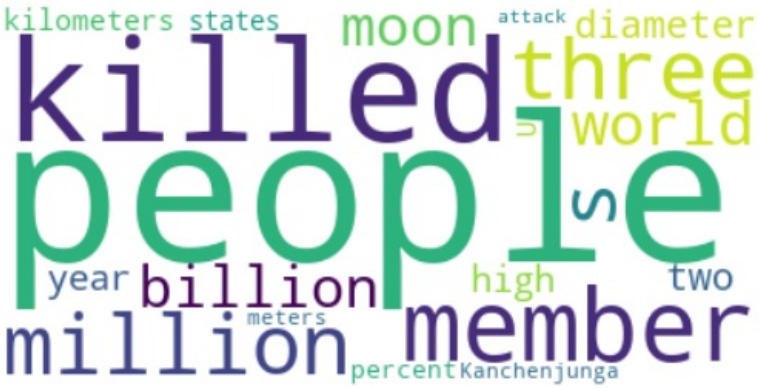
In [80]:
```
def generate_word_cloud(text):
    wordcloud = WordCloud(stopwords=STOPWORDS,
    background_color='white',
    collocations=False,
    max_words=20).generate(text)

    # Display the generated image:
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis("off")
```

```
    plt.show()
```

RTE_Quant

In [81]:
```
premise_list = list(rte["premise"].values)
hypothesis_list = list(rte["hypothesis"].values)
premise, hypothesis = " ".join(premise_list), " ".join(hypothesis_list)
# Create and generate a word cloud image:
generate_word_cloud(premise)
```

In [82]:
```
generate_word_cloud(hypothesis)
```



Based on the most frequent unigrams in this dataset, RTE_Quant seems to be based on sentences extracted from news articles. We notice that 2 and 3 are very frequent quantities, alongside the "million" word.

RedditNLI

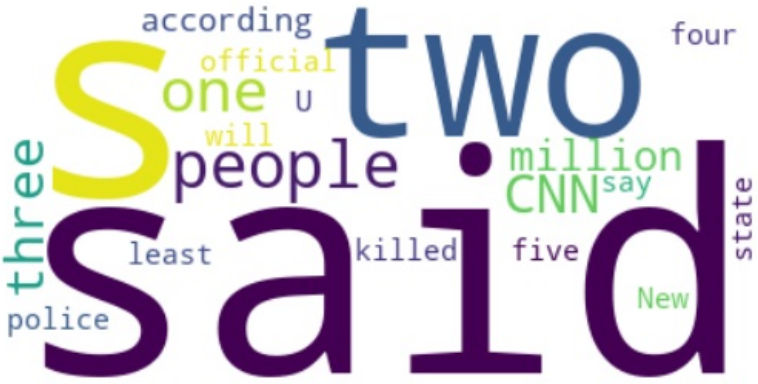
In [83]:
```
premise_list = list(reddit["premise"].values)
hypothesis_list = list(reddit["hypothesis"].values)
premise, hypothesis = " ".join(premise_list), " ".join(hypothesis_list)
# Create and generate a word cloud image:
generate_word_cloud(premise)
```

In [84]:
```
generate_word_cloud(hypothesis)
```

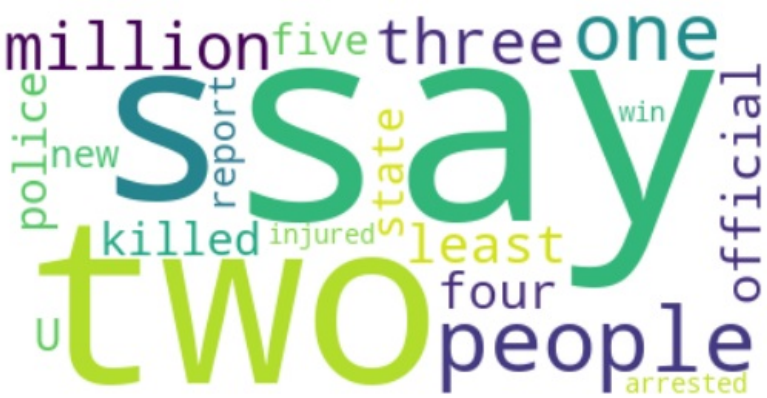
The RedditNLI dataset, containing sentences extracted from financial/economic headlines on Reddit, has among the most frequent unigrams words related to the financial sector. We notice that there are no quantities among the most frequent unigrams, in contrast with other datasets. This may be because in finance, especially when we talk about stock prices, the number are usually very specific.

NewsNLI

In [85]:
```
premise_list = list(news["premise"].values)
hypothesis_list = list(news["hypothesis"].values)
premise, hypothesis = " ".join(premise_list), " ".join(hypothesis_list)
# Create and generate a word cloud image:
generate_word_cloud(premise)
```



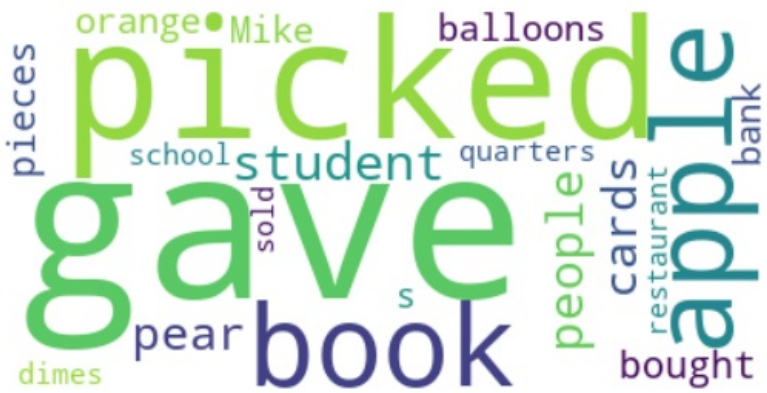
In [86]:
```
generate_word_cloud(hypothesis)
```



For the NewsNLI dataset, we observe a combination of common quantities (one, two, three four, million) and words likely to be used in a news article (people, police, state, say, said, killed etc).

AWPNLI

In [87]:
```
premise_list = list(awp["premise"].values)
hypothesis_list = list(awp["hypothesis"].values)
premise, hypothesis = " ".join(premise_list), " ".join(hypothesis_list)
# Create and generate a word cloud image:
generate_word_cloud(premise)
```

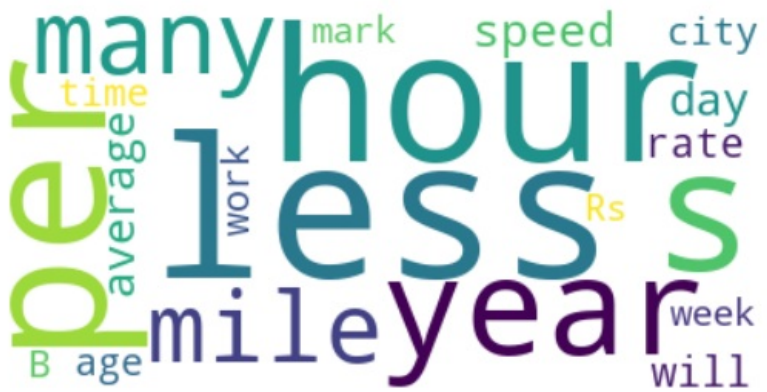


In [88]:
```
generate_word_cloud(hypothesis)
```

As AWPNLI is a dataset based on Math word problems, the most frequent words inside it are nouns usuallu used as entities in this type of exercises (apples, book, orange). We also observe words which indicate operations like addition or subtraction (left, needed, gave, picked, bought).

StressTest

In [89]:
```
premise_list = list(stress["premise"].values)
hypothesis_list = list(stress["hypothesis"].values)
premise, hypothesis = " ".join(premise_list), " ".join(hypothesis_list)
# Create and generate a word cloud image:
generate_word_cloud(premise)
```



In [90]:
```
generate_word_cloud(hypothesis)
```

Although the StressTest is also based on Math word problems, we notice a discrepancy between the word cloud for this set and those for AWPNLI. The StressTest seems to focus on problems with topics such as time (hour, day, age) and distance (mile, city, speed).

Let's take a look at what type of textual quantifiers are in the datasets.

In [91]:
```
# List of most common quantifier (this list is not exhaustive)
quantifiers = ["at least", "not less than", "no less than", "minimum of", "equal to or greater than", "no fewer
"greater than or equal to", "not fewer than", "down to", "less than", "below", "under", "lower than", "above",


def count_lookup_phrases(sentences):
    total_cnt, premise_count = 0, 0
    phrases_found = set()
    for phrase in quantifiers:
        for premise in sentences:
            premise_counted = False
            if phrase in premise:
                phrases_found.add(phrase)
                total_cnt += 1
                if not premise_counted:
                    premise_count += 1
    return total_cnt, premise_count, phrases_found
```

AWPNLI

In [92]:
```
print(count_lookup_phrases(list(awp["premise"].values)))
```

```
(18, 18, {'up to', 'over', 'around', 'about'})
```

In [93]:
```
print(count_lookup_phrases(list(awp["hypothesis"].values)))
```

```
(8, 8, {'over'})
```

Only few of the AWPNLI samples have quantifiers. The focus of this dataset is on the model understanding it has to do calculations based on the quantities in the premise, so he can infer a quantity that must be compared to the one in the hypothesis. Moreover, given the synthetic source of the dataset (math word problems), the lack of diversity in the user quantifiers is also understandable.

RedditNLI

In [94]:
```
print(count_lookup_phrases(list(reddit["premise"].values)))
```

```
(41, 41, {'above', 'around', 'near', 'close to', 'under', 'at least', 'about', 'more than', 'down to', 'exceedin
g', 'over', 'below'})
```

In [95]:
```
print(count_lookup_phrases(list(reddit["hypothesis"].values)))
```

```
(35, 35, {'above', 'around', 'near', 'under', 'at least', 'about', 'more than', 'down to', 'up to', 'over', 'bel
ow'})
```

Also for RedditNLI, a small fraction of the samples contain quantifiers. The focus in this dataset is on direct comparisons of the quantities and understanding if the quantities in the hypothesis are related to those in the premise and if they can be inferred or not.

NewsNLI

In [96]:
```
print(count_lookup_phrases(list(news["premise"].values)))
```

```
(296, 296, {'above', 'around', 'near', 'close to', 'under', 'at least', 'about', 'more than', 'down to', 'approx
imately', 'as high as', 'roughly', 'up to', 'over', 'below', 'less than'})
```

In [97]:
```
print(count_lookup_phrases(list(news["hypothesis"].values)))
```

```
(170, 170, {'above', 'no more than', 'around', 'near', 'under', 'close to', 'at least', 'about', 'more than', 'd
own to', 'up to', 'over', 'below', 'less than'})
```

The NewsNLI dataset is more abundant in quantifiers. Given the nature of this dataset, namely news articles, the presence of quantifiers and their diversity is expected, since they play a key role in highlighting ideas and summarizing information, as well as making information easier to understand and/or remember (i.e. think of reporting the number "2473" compared to "at least 2400")

RTE_Quant

In [98]:
```
print(count_lookup_phrases(list(rte["premise"].values)))
```

```
(72, 72, {'above', 'around', 'near', 'under', 'at least', 'about', 'more than', 'greater than', 'roughly', 'up t
o', 'over', 'below', 'less than'})
```

In [99]:
```
print(count_lookup_phrases(list(rte["hypothesis"].values)))
```

```
(25, 25, {'near', 'at least', 'about', 'more than', 'greater than', 'roughly', 'up to', 'over', 'less than'})
```

This dataset contains a relatively high ratio of samples with quantifiers (at least in the premise). Given the natural language source of this dataset, the presence of quantifiers is expected.

StressTest

In [100…
```
print(count_lookup_phrases(list(stress["premise"].values)))
```

```
(3122, 3122, {'above', 'no more than', 'around', 'near', 'under', 'at least', 'maximum of', 'more than', 'about'
, 'approximately', 'greater than', 'minimum of', 'up to', 'at most', 'over', 'below', 'less than'})
```

In [101…
```
print(count_lookup_phrases(list(stress["hypothesis"].values)))
```

```
(4368, 4368, {'above', 'no more than', 'around', 'near', 'under', 'at least', 'maximum of', 'more than', 'about'
, 'approximately', 'greater than', 'minimum of', 'up to', 'at most', 'over', 'below', 'less than'})
```

This dataset contains a large fraction of samples with quantifiers. Given that this set is obtained from algebra word problems, the use of quantifiers and their diversity makes sense. The used quantifiers are basic ones, often encountered in math sentences.

Finally, let's identify the baselines for each dataset, as well as a baseline for EQUATE as a whole

In [102…
```
datasets = [rte, news, reddit, awp, stress]
dataset_names = ["RTE_Quant", "NewsNLI", "RedditNLI", "AWPNLI", "StressTest"]

for dataset_df, dataset_name in zip(datasets, dataset_names):
    print(f"###############\n{dataset_name}")
    label_frequency = dataset_df['label'].value_counts(normalize=True).reset_index()  # find fraction of sample
    baseline_ratio = label_frequency['proportion'].max()
    label = label_frequency[label_frequency['proportion'] == baseline_ratio].iloc[0]["label"]
    print(f"Baseline: {round(baseline_ratio, 4)} (label: {label})")
```

```
###############
RTE_Quant
Baseline: 0.5783 (label: neutral)
###############
NewsNLI
Baseline: 0.5072 (label: entailment)
###############
RedditNLI
Baseline: 0.584 (label: entailment)
###############
AWPNLI
Baseline: 0.5 (label: entailment)
###############
StressTest
Baseline: 0.3333 (label: entailment)
```

In [103…
```
equate_df = pd.DataFrame()

for dataset in datasets:
    equate_df = pd.concat([equate_df, dataset], ignore_index=True)

print(equate_df.shape)
```

```
(9702, 30)
```

In [104…
```
label_frequency = equate_df['label'].value_counts(normalize=True).reset_index()  # find fraction of samples ass
baseline_ratio = label_frequency['proportion'].max()
label = label_frequency[label_frequency['proportion'] == baseline_ratio].iloc[0]["label"]
print(f"Baseline: {round(baseline_ratio, 4)} (label: {label})")
```

```
Baseline: 0.3711 (label: entailment)
```