An Empirical Study of Methods for Matching Factual Claims

Anonymous NAACL submission

Abstract

In this study, we investigated the efficacy of using natural language inference (NLI) techniques for claim matching, which is an important task in the workflow of an end-to-end computational fact-checking system. Given a repository of known fact-checks produced by professionals and a factual claim fed from an upstream claim collecting/monitoring component of such a system, the goal of claim matching is to identify those known fact-checks that match the claim and furthermore the relationship between the claim and each matched fact-check. Such relationships include paraphrasing, entailment, contradiction, as well as generally speaking the claim and the fact-check being textually and semantically similar. The matched fact-checks are helpful to fact-checkers in finding references and even automating fact-checking by providing instant verdicts on certain matched claims. In the paper we report the results of an empirical evaluation of a variety of algorithms for claim matching, ranging from major entailment decision algorithms to an adaptation of Google's Transformer, a self-attention based neural network model. Furthermore, while there are datasets for evaluating general-purpose NLI algorithms, there is not such a dataset specifically designed for matching factual claims. Hence, we created a benchmark dataset for this study, which can become a valuable resource to the community of researchers and practitioners in related fields of study.

1 Introduction

Our society is struggling with an unprecedented amount of falsehood that is harming wealth, democracy, health, and national security. In domestic political controversies, politicians repeat false claims even after they are debunked. "Fake news" is fabricated to spread derogatory rumors, promote societal and political tensions, manipulate public opinion, and influence national election outcome. The problem is further exacerbated by social media bots and clickbaits that spread and amplify falsehoods.

Professional fact-checkers take on the hard battle to counter misinformation and disinformation. According to the Duke Reporters' Lab, [1] the number of active fact-checking organizations, including the likes of The Washington Post, New York Times, and FactCheck.org, has grown from 44 in 2014 to 162 in December 2018. Fact-checkers vet claims by presenting relevant data and documents and publishing their verdicts. For instance, PolitiFact.com, one of the earliest and most popular fact-checking projects, gives factual claims truthfulness ratings such as true, half true, false, and even "pants on fire".

However, the insurmountable amount of false information is way beyond the capability of current fact-checkers to keep up with. The intellectually demanding and laborious process of fact-checking takes the professionals about one day to research and write a typical article about a factual claim (Hassan et al., 2015a). This difficulty leaves many harmful claims unchecked. Even if a fact-check has already been published, it is mostly unrealistic to expect the public to relate an existing fact-check with a claim they just encounter.

In response to these challenges, while the initial call to arms to research on computational fact-checking was made nearly a decade ago (Cohen et al., 2011), the last several years have witnessed a substantial growth in interests and efforts in this arena. These efforts tackle various fronts, from detecting important factual claims that are worth checking (Hassan et al., 2015b; Gencheva et al., 2017; Jimenez and Li, 2018; Konstantinovskiy et al., 2018), to using databases for discerning fac-

[1] https://reporterslab.org/fact-checking/

tual claims' robustness (Wu et al., 2014, 2017) and truthfulness (Ciampaglia et al., 2015; Shi and Weninger, 2016; Jo et al., 2018), to building end-to-end fact-checking systems (Babakar and Moy, 2016; Hassan et al., 2017a,b).

This paper presents the results of an empirical study of using computational methods to assist *claim matching*, an important step in the workflow of fact-checking. Given a factual claim, this step aims at matching the claim against a repository of existing fact-checks. The premise is that politicians and public figures keep making the same false claims. While politicians may refrain themselves from making outright false claims to avoid being fact-checked, oftentimes they even double down after their false claims are debunked. [2]

In the simplest case, when someone repeats a claim identical to one that has been fact-checked, the existing fact-check's verdict can be presented to fact-checkers for further investigation and directly to news audience as well (Graves, 2018). An example of the former case is Full Fact's in-house platform. [3] An example of the latter case is the pop-up fact-checking app FactStream produced by the Duke Reporters' Lab. These approaches can all leverage the tens of thousands of fact-checks produced by various organizations over the years. These fact-checks are now organized into repositories such as the Google Fact Check Explorer, [4] using annotation tools such as the Schema.org based ClaimReview markup tool [5] and the "Share the Facts" gadget. [6]

However, claim matching is not always as straightforward as finding existing fact-checks on identical claims. Instead, oftentimes claims are rephrased, a fact-check partially supporting or refuting a claim at hand is still insightful in vetting the claim, and even a related fact-check can be helpful. For instance, consider the following two real-world statements:

S1: *"African-Americans are more likely to be arrested by police and sentenced to longer prison terms for doing the same thing that whites do."*

S2: *"... if you're a young African-American man and you do the same thing as a young white man, you are more likely to be arrested, charged, convicted, and incarcerated."*

[2] https://wapo.st/2rucTq8
[3] https://fullfact.org/
[4] https://toolbox.google.com/factcheck/explorer
[5] https://schema.org/ClaimReview
[6] http://www.sharethefacts.org/

Albeit lacking an exact match, these two statements express the same idea and thus the verdict on one can effectively help fact-check the other. Hence, a more general approach to claim matching is to detect rephrased claims, partial matches, and even related claims. Such an approach can benefit from a wide range of natural language processing techniques, including coreference resolution (Clark and Manning, 2015), entity matching (Mudgal et al., 2018), paraphrase detection (Socher et al., 2011), semantic similarity (Ji and Eisenstein, 2013), and textual entailment (Androutsopoulos and Malakasiotis, 2010).

In this paper, we empirically examined the feasibility of using *textual entailment* techniques for claim matching. The goal of textual entailment, also known as natural language inference (NLI), is to determine whether a hypothesis statement can be inferred from a premise statement. While this standard notion of textual entailment implies a directional relation, our study generalizes it in two ways. First, since our goal is to detect matching claims, we consider entailment in both directions of inference, i.e., two statements S1 and S2 match if S1 entails S2 and/or S2 entails S1. Second, in textual entailment "contradiction" (i.e., the premise statement can falsify the hypothesis statement) is a relation opposite to "entailment", but "entailment" and "contradiction" are both useful in finding matching claims for fact-checking purpose. Our work is a novel adaptation and extension of textual entailment techniques, as a recent survey (Cazalens et al., 2018) stated that "textual entailment has never been applied explicitly to fact checking problems." Our empirical study also compares NLI-based methods with an IR-based method and a transformation-based method. Another contribution made in the paper is the creation of a sizable benchmark dataset specifically for claim matching.

We aim at starting a line of investigation into an important fact-checking task. Adapting and extending textual entailment methods is only an initial step along this direction. Furthermore, the problem may not be limited to only matching individual claims. For instance, (Wang et al., 2018) studied how to find relevant, supporting web pages for a given claim and further for fact-checking articles featuring one or more claims.

2 Methods for Matching Factual Claims

2.1 Methods Based on Textual Entailment

Textual entailment is a directional relation between two text fragments—the premise *P* and the hypothesis *H*—such that a human with a common sense can infer that *H* is most likely true on the basis of the content of *P* (Giampiccolo et al., 2007). Consequently, the entailment recognition task is to decide if *P* entails *H*. The steady progress in textual entailment over a decade has made it a well-established natural language processing benchmark task. Especially, the PASCAL Recognizing Textual Entailment (RTE) Challenges (Dagan et al., 2006) encouraged the development of numerous systems. One such system, the EXCITEMENT Open Platform (EOP), [7] provides a battery of different top-performing algorithms, linguistic analysis components, as well as a large number of knowledge resources for experimenting with new approaches. Our empirical study assesses four representative algorithms in it, namely EditDistance (Kouylekov and Magnini, 2005), TIE (Wang and Neumann, 2007), AdArte (Zanoli and Colombo, 2017), and P1EDA (Noh et al., 2015).

EditDistance: This approach performs a sequence of editing operations (insertion, deletion, and substitution) to convert the premise *P* into the hypothesis *P*. EOP provides two variants of this approach. While EditDistanceEDA uses a pre-defined constant cost for each editing operation, EditDistancePSOEDA automatically calculates each operation's cost using particle swarm optimization (Kennedy and Eberhart, 1995).

TIE: This algorithm extracts structural features from the syntactic dependency trees of *P* and *H* to train a classification model. The features are in the form of triples $\langle node_1, rel, node_2 \rangle$, in which $node_1$ is the head, rel is the dependency relation, and $node_2$ is the modifier. A function measures the similarity between *P* and *H* by overlapping triples in features extracted from them. The similarity measures are input features to a *subsequence kernel method* for discerning the entailment relation.

AdArte: This approach resembles a combination of EditDistance and TIE. Similar to TIE, it employs dependency trees to represent statements. Similar to EditDistance, it calculates the editing operations required for transforming *P* into *H*. However, the editing operations are performed on the dependency trees instead of token fragments. Additionally, it utilizes knowledge resources such as WordNet[8] to recognize whether different textual expressions can preserve entailment (synonyms and hypernyms) or not (antonyms). The editing operations along with such information regarding entailment preservation are used as the input features to an entailment classifier based on support vector machine.

P1EDA: This is also a supervised classification model, using two types of features. The first type of features comes from *alignment links* which record how different pairs of tokens/phrases from *P* and *H* align with each other. These alignment links are detected by three different aligners. The lexical aligner creates an alignment link if there is a semantic relationship between two tokens/phrases according to lexical resources. The paraphrase aligner defines an alignment link if the tokens/phrases are paraphrases. The lemma identity aligner produces an alignment link if the tokens/phrases are reduced to the same lemmas. The second type of features includes the ratios of overlaps between 1) all words, 2) all content words, 3) verbs, and 4) proper names in *P* and *H*. These two types of features form an input feature vector to a supervised classification algorithm.

2.2 Baseline Methods

In this study we also considered two baseline methods for claim matching. The *IR-based model* uses part-of-speech tagging and semantic similarity to generate features. After removing stopwords, every word of each sentence is tagged with a part-of-speech. All words which are not verbs, adjectives, adverbs, or nouns are then removed, since such words contribute little to a sentence's meaning. Next, a score is calculated for each of the four relevant parts of speech. For a given part-of-speech, the sentence with fewer words of this type is found. Then, every word of that part-of-speech in that sentence is compared against all words of the same part-of-speech in the other sentence to find the pair with highest Wu-Palmer semantic similarity (Wu and Palmer, 1994). These maximum similarities are summed and normalized using the harmonic mean. The harmonic mean guarantees scores between 0 and 1 and incen-

[7] https://github.com/hltfbk/EOP-1.2.3/wiki

[8] https://wordnet.princeton.edu/

tivizes sentences with similar numbers of words of each part-of-speech. Finally, the noun, verb, adverb, and adjective scores are used as input features for a support vector machine using radial-basis function kernel.

The *transformer-based model* is adapted from a model developed by OpenAI (Radford et al., 2018). It aims to tackle two challenges that have plagued traditional natural language inference (NLI) approaches. The first challenge, the difficulty of identifying word dependencies, is addressed by the transformer framework (Vaswani et al., 2017), a type of neural network created by Google. The transformer framework improves upon previous architectures (CNN, RNN, RNN with LSTM) by using attention mechanisms which determine meaning based on a weighted average of dependency on all other words in the sentence. The second challenge, the limited quantity of labeled NLI data, is mitigated by an unsupervised preprocessing step using the BooksCorpus dataset (Zhu et al., 2015) of over 7,000 unpublished books. OpenAI trained a multi-layer transformer decoder to predict the next token of a text given the immediately preceding tokens.

3 Datasets

All the methods considered in Section 2 use supervised classification models. In this section we explain the training and test datasets used in our study.

3.1 Training Set

We used two different datasets, RTE-3 and MNLI, for training claim-matching models.

RTE-3 (Giampiccolo et al., 2007) is the benchmark dataset used by the textual entailment community in their third annual challenge. It consists of 1,600 (premise sentence, hypothesis sentence) pairs. Each pair is labeled as either "entailment" or "non-entailment". It is a balanced dataset with slightly more non-entailment pairs than entailment ones.

The Multi-Genre Natural Language Inference (MNLI) training dataset contains 392,702 sentence pairs from 6 written and spoken genres, including government press releases and telephone conversations (Williams et al., 2018). The pairs are evenly distributed in three classes which are "entailment", "contradiction", and "neutral" (i.e., "not matching"). For the purposes of claim matching the "entailment" and "contradiction" classes were combined into one class "matching". As explained in Section 1, both "entailment" and "contradiction" are useful in finding matching claims for fact-checking purpose.

3.2 Test Set

For evaluating the classification models' performance, we generated ClaimPairs, a test set consisting of 1,111 (premise claim, hypothesis claim) pairs. The premise claims are the subjects of fact-checks in PolitiFact, and the hypothesis claims are from the 2016 U.S. Primary and General Election debate transcripts. More specifically, we used a repository of fact-checks hosted by the Duke Reporters' Lab and retrieved 3,375 fact-checks published by PolitiFact in the past. The debate transcripts have over 32,000 sentences. We used the ClaimBuster API (Hassan et al., 2017a,b) to find factual claims among these sentences and removed those with only four or fewer words. This led to around 6,000 hypothesis claims. We created a cross product of the premise and hypothesis claims, resulting in about 20 million pairs. We further removed stop words from the claims and obtained the lemma of each word. After that, we removed identical pairs as well as pairs with identical lemmas. Furthermore, all pairs with less than six common lemmas (which are the easy cases) were filtered out. The threshold six was selected empirically—a smaller value led to a much bigger dataset which is difficult for human coders to label. The final dataset is comprised of 1,111 pairs of premise and hypothesis claims.

Each (premise claim, hypothesis claim) pair in ClaimPairs is labeled into one of three classes, as follows. "Entailment", if a human can imply the hypothesis from the premise claim. "Contradiction", if the premise claim falsifies the hypothesis claim. We employed the methods in (Marneffe et al., 2008) to identify contradictions based on the presence of negation, antonyms, numeric mismatch, and so on. "Neutral", if a human can neither infer nor falsify the hypothesis claim based on the premise claim. As mentioned earlier, both "entailment" and "contradiction" are useful in finding matching claims for fact-checking purpose.

As also explained in Section 1, for detecting matching claims, we consider entailment in both directions of inference, i.e., two statements S1 and S2 match if S1 entails S2 and/or S2 entails

ID	Premise Claim (P)	Hypothesis Claim (H)	P $\rightarrow$ H	$P \leftarrow$ H
1	Hostages were released as soon as Ronald Reagan took office because Iran perceived that America was "no longer under the command of someone weak."	It's worth remembering Iran released our hostages the day Ronald Reagan was sworn into office.	Entailment	Neutral
2	While our people work longer hours for lower wages, almost all new income goes to the top 1 percent.	I'm running for president because our economy is rigged because working people are working longer hours for lower wages and almost all of new wealth and income being created is going to the top one percent.	Neutral	Entailment
3	Forty percent of illegal immigrants are "people coming legally on visas and overstaying their visas."	40 percent of the people who come here illegally come legally, and then they overstay the visa.	Entailment	Entailment
4	When President Obama took office, we were losing 700,000 jobs a month. Now we've had job growth, I think, for 24 consecutive months.	When President Obama came into office we were losing 800,000 jobs a month, 800,000 jobs a month.	Contradiction	Neutral
5	I balanced the budget for four straight years, paid off $405 billion in debt.	And I've done it when I was in Washington when we had a balanced budget; had four years of balanced budgets; paid down a half-trillion of debt.	Neutral	Contradiction
6	New Jersey has had "seven credit downgrades" since Chris Christie became governor.	Under Chris Christie's governorship of New Jersey, they've been downgraded nine times in their credit rating.	Contradiction	Contradiction
7	Rep. Paul Ryan's budget proposal cuts "nothing" from Medicare, Social Security or defense in the next two to three years, and "in three years, he does not cut one dime from the debt."	Senator Paul, the budget deal crafted by Speaker Boehner and passed by the House today makes cuts in entitlement programs, Medicare and Social Security disability, which are the very programs conservatives say need cutting to shrink government and solve our country's long-term budget deficit.	Neutral	Neutral

Table 1: Sample (premise claim, hypothesis claim) pairs.

	ClaimPairs	ClaimPairs-R
Entailment	119	125
Contradiction	22	47
Neutral	970	939

Table 2: Distribution of "entailment", "contradiction" and "neutral" classes in ClaimPairs and ClaimPairs-R.

S1. For this reason, we created another dataset ClaimPairs-R by swapping the premise and the hypothesis claims in ClaimPairs, and we labeled ClaimPairs-R by following the aforementioned approach. Table 1 lists some sample pairs along with their labels for both directions. Note that the labels in these two directions do not necessarily form a reverse relation.

Table 2 shows the distribution of classes for ClaimPairs and ClaimPairs-R datasets. Both datasets are imbalanced with 87.3% and 84.5% "neutral" pairs, respectively. Since the majority of both datasets are "neutral", we used them only as test sets but not as training sets.

4 Evaluation Results

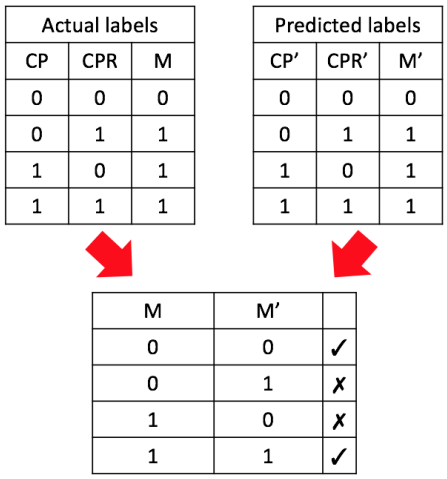

Figure 1: Evaluation metric for bi-directional entailment. CP denotes the ClaimPairs dataset and CPR denotes the ClaimPairs-R dataset. 0 means "non-entailment" while 1 is for "entailment".

We conducted two sets of evaluations. In the first set of evaluations, we measured the perfor-

Algorithms	p_En	p_Non	r_En	r_Non	f1_En	f1_Non
EditDistance (Lemma)	0.49	**0.97**	**0.81**	0.88	0.61	0.92
EditDistance (Lemma+WN)	0.54	0.96	0.77	0.91	**0.64**	0.93
EditDistance (Token)	0.51	0.96	0.74	0.90	0.61	0.93
EditDistance (Token+WN)	0.53	0.96	0.74	0.91	0.62	0.93
PSO EditDistance (Token)	0.48	0.97	0.79	0.88	0.60	0.92
PSO EditDistance (Token+WN)	0.54	0.96	0.74	0.91	0.63	0.93
TIE (OpenNLP)	**0.96**	0.92	0.38	**1.00**	0.54	**0.96**
TIE (VO+TP+TPPos+TS)	0.83	0.90	0.24	0.99	0.37	0.94
TIE (WN+TP+TPPos+TS)	0.82	0.90	0.23	0.99	0.36	0.94
TIE (WN+VO)	0.47	0.93	0.55	0.91	0.51	0.92
TIE (WN+VO+TP+TPPos)	0.83	0.91	0.31	0.99	0.45	0.95
TIE (WN+VO+TP+TPPos+TS)	0.83	0.91	0.31	0.99	0.45	0.95
TIE (WN+VO+TS)	0.76	0.90	0.23	0.99	0.35	0.94
AdArte	0.35	0.95	0.70	0.81	0.47	0.88
P1EDA	0.61	0.95	0.69	0.94	**0.64**	0.94
IR (RTE-3)	0.50	0.94	0.58	0.92	0.54	0.93
IR (MNLI)	0.50	0.95	0.67	0.90	0.57	0.93
Transformer (MNLI)	0.41	**0.97**	**0.81**	0.83	0.55	0.90

Table 3: Results of each model's performance for the claim matching task, in terms of Precision (p), Recall (r), and F-measure (f1). The models' performance is measured by accuracy of one direction of entailment on the ClaimPairs dataset. The two classes are "matching" (i.e., "entailment" or "contradiction", denoted as En) and "not matching" (i.e., "non-entailment" or "neutral", denoted as NonEn).

Algorithms	p_En	p_Non	r_En	r_Non	f1_En	f1_Non
EditDistance (Lemma)	0.57	**0.94**	0.85	0.78	0.68	0.85
EditDistance (Lemma+WN)	0.63	**0.94**	0.85	0.83	**0.72**	0.88
EditDistance (Token)	0.62	0.93	0.81	0.83	0.71	0.88
EditDistance (Token+WN)	0.61	**0.94**	0.83	0.82	0.71	0.87
PSO EditDistance (Token)	0.59	0.93	0.83	0.80	0.69	0.86
PSO EditDistance (Token+WN)	0.61	**0.94**	0.83	0.82	0.71	0.87
TIE (OpenNLP)	**0.90**	0.84	0.45	0.98	0.60	0.90
TIE (VO+TP+TPPos+TS)	0.83	0.80	0.29	0.98	0.43	0.88
TIE (WN+TP+TPPos+TS)	0.84	0.80	0.27	0.98	0.41	0.88
TIE (WN+VO)	0.89	0.86	0.55	0.98	0.68	**0.92**
TIE (WN+VO+TP+TPPos)	0.88	0.80	0.30	**0.99**	0.45	0.89
TIE (WN+VO+TP+TPPos+TS)	0.88	0.80	0.30	**0.99**	0.45	0.89
TIE (WN+VO+TS)	0.79	0.80	0.29	0.97	0.42	0.88
AdArte	0.45	0.90	0.77	0.68	0.57	0.77
P1EDA	0.66	0.88	0.66	0.88	0.66	0.88
IR (RTE-3)	0.47	0.87	0.68	0.74	0.56	0.80
IR (MNLI)	0.51	0.90	0.77	0.75	0.61	0.82
Transformer (MNLI)	0.45	**0.94**	**0.87**	0.63	0.59	0.75

Table 4: Results of each model's performance for the claim matching task, in terms of Precision (p), Recall (r), and F-measure (f1). The models' performance is measured by accuracy of entailment in both directions on the ClaimPairs and ClaimPairs-R datasets.

mance of various methods in terms of accuracy of one-direction entailment (from premise to hypothesis) on the ClaimPairs dataset. Table 3 depicts the results of each model's performance for the claim matching task in terms of precision, recall and f1-score per class. In the second set of evaluations, we used both ClaimPairs and ClaimPairs-R. For a pair (P,H) in ClaimPairs and its corresponding pair (H,P) in ClaimPairs-R. With regard to ground-truth, if either P matches H according to the label in ClaimPairs (recall that "matching" means either "entail" or "contradict") or H matches P according to ClaimPairs-R, we consider P and H "matching". With regard to prediction, if either P matches H or H matches P based on the prediction by a method being evaluated, we consider the prediction to be "matching" If the ground-truth and the prediction are consistent, the method is considered correct on the pair. This scheme of evaluation is depicted in Figure 1.

Methods from EOP: We evaluated the various implementations of textual entailment techniques provided by the EOP system. In our results Tables 3 and 4, we denoted WordNet:lexical database

Premise	Hypothesis
Independent analysts "found that Trump's tax plan, given to the wealthy and big corporations, would rack up $30 trillion in debt.	Independent experts have looked at what I've proposed and looked at what Donald's proposed, and basically they've said this, that if his tax plan, which would blow up the debt by over $5 trillion and would in some instances disadvantage middle-class families compared to the wealthy, were to go into effect, we would lose 3.5 million jobs and maybe have another recession.
His free public university tuition program "is paid for by a tax on Wall Street's speculation."	That is why I'm going to have a tax on Wall Street speculation to make certain that public colleges and universities in America are tuition free.
A black male baby born today, if we do not change the system, stands a one-in-three chance (of) ending up in jail.	As a black man in America, if I were born today I'd have a one in three chance of ending up in prison in my life.
President Obama has broken his pledge to the American people to be transparent throughout (health care reform negotiations).	As we learned with President Obama's broken promise that everyone could keep their plan, any major plan – change in health care policy carries with it the risk that some people will lose their insurance coverage or have to change it.
As a candidate, President Obama "declared that everyone deserves access to reproductive health care that includes abortion, and vowed that this 'right' would be at the heart of his health care reform plan if elected president."	As we learned with President Obama's broken promise that everyone could keep their plan, any major plan – change in health care policy carries with it the risk that some people will lose their insurance coverage or have to change it.
We have 17 intelligence agencies, civilian and military, who have all concluded that these espionage attacks, these cyberattacks, come from the highest levels of the Kremlin, and they are designed to influence our election."	This has come from the highest levels of the Russian government, clearly, from Putin himself, in an effort, as 17 of our intelligence agencies have confirmed, to influence our election.
Hillary Clinton "promised, running for the Senate years ago, 200,000 jobs for upstate New York. ... Not only didn't they come, but they lost so many jobs."	I heard them when they were running for the Senate in New York, where Hillary was going to bring back jobs to upstate New York and she failed.

Table 5: Hard cases where all claim matching models failed to classify them correctly.

as WN, VerbOcean:semantic network of verbs as VO, TreePattern:tree mining algorithm as TP, TreePattern with POS tags as TPPos, TreeSkeleton:tree mining algorithm as TS. For experiments using EOP-based methods, we used RTE-3 dataset (1,600 pairs) for training.

IR-based model: We evaluated the IR-based model trained on RTE-3 and MNLI, respectively, using default hyperparameters. The model trained on MNLI attained slightly better F-measure than the model on RTE-3, using ClaimPairs as the test set in both cases.

Transformer-based model: We used OpenAI's weights from the unsupervised step as a starting point to train the Transformer model on the MNLI dataset. Very little preprocessing is done, and input sentences are represented as concatenations of the GloVe embeddings of their words. Due to computational constraints, default hyperparameters were utilized and the model was only trained on MNLI. The Transformer model displayed a strong tendency towards positively classifying instances, likely due to the class imbalance in the MNLI dataset.

Almost all TIE models have the best performance in terms of recall and f1-score for NonEntailment class, but they and suffer for the recall of Entailment. However, TIE (WN+VO) perform considerably better than other TIE models for Entailment recall. We can say that TP, TS, and TPPos do not contribute to the classification of Entailment. On the contrary, the lack of these knowledge resources leads a better recall and f1-score.

Our intuition when we decided to evaluate the algorithms' performance independently of the direction was that would perform better on recognizing entailment. As it seems for entailment pairs, outperform the one-way relation in almost all cases, while for the non-entailment relations did not perform as well. Surprising to our beliefs EOP methods perform better compared to most recent methods of the IR-based model, and Transformer-based model.

Additionally, Table 5 shows all hard cases where none of the models was able to make correct predication.

5 Conclusion

In our study, we shed some light at the adaptation of textual entailment techniques in the workflow of

the fact-checking. We presented how claim matching can benefit from available systems by conducting various experiments. Moreover, we generated a test-bed dataset specifically created for the claim matching task.

References

Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187.

Mevan Babakar and Will Moy. 2016. The state of automated factchecking. *Full Fact 28*.

Sylvie Cazalens, Philippe Lamarre, Julien Leblay, Ioana Manolescu, and Xavier Tannier. 2018. A content management perspective on fact-checking. In *"Journalism, Misinformation and Fact Checking" alternate paper track of" The Web Conference"*, pages 1–10.

Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M. Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational fact checking from knowledge networks. *PLOS ONE*, 10:1–13.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415. Association for Computational Linguistics.

Sarah Cohen, Chengkai Li, Jun Yang, and Cong Yu. 2011. Computational journalism: A call to arms to database researchers. In *CIDR*, pages 148–151.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine learning challenges. Evaluating predictive uncertainty, visual object classification, and recognising textual entailment*, pages 177–190. Springer.

Pepa Gencheva, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, and Ivan Koychev. 2017. A context-aware approach for detecting worth-checking claims in political debates. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 267–276.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9. Association for Computational Linguistics.

Lucas Graves. 2018. Factsheet: Understanding the promise and limits of automated fact-checking. Technical report, Tech. Rep.). Reuters Institute for the Study of Journalism, University of Oxford.

Naeemul Hassan, Bill Adair, James T. Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. 2015a. The quest to automate fact-checking. In *2015 Computation+Journalism Symposium*.

Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017a. Toward automated fact-checking: Detecting check-worthy factual claims by ClaimBuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1803–1812. ACM.

Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015b. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM)*, pages 1835–1838.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li, and Mark Tremayne. 2017b. ClaimBuster: The first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10(12):1945–1948.

Yangfeng Ji and Jacob Eisenstein. 2013. Discriminative improvements to distributional sentence similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 891–896.

Damian Jimenez and Chengkai Li. 2018. An empirical study on identifying sentences with salient factual statements. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Saehan Jo, Immanuel Trummer, Weicheng Yu, Daniel Liu, and Niyati Mehta. 2018. The factchecker: Verifying text summaries of relational data sets.

J. Kennedy and R. Eberhart. 1995. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4.

Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2018. Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. *Computing Research Repository*, arXiv:1809.08193.

Milen Kouylekov and Bernardo Magnini. 2005. Recognizing textual entailment with tree edit distance algorithms. In *Proceedings of the First Challenge Workshop Recognising Textual Entailment*, pages 17–20.

Marie-Catherine Marneffe, Anna N Rafferty, and Christopher D Manning. 2008. Finding contradictions in text. pages 1039–1047.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*, pages 19–34. ACM.

Tae-Gil Noh, Sebastian Padó, Vered Shwartz, Ido Dagan, Vivi Nastase, Kathrin Eichler, Lili Kotlerman, and Meni Adler. 2015. Multi-level alignments as an extensible representation basis for textual entailment algorithms. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 193–198.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Baoxu Shi and Tim Weninger. 2016. Discriminative predicate path mining for fact checking in knowledge graphs. *Knowledge-Based Systems*, 104(C):123–133.

Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in neural information processing systems*, pages 801–809.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Rui Wang and Günter Neumann. 2007. Recognizing textual entailment using a subsequence kernel method. In *AAAI*, volume 7, pages 937–945.

Xuezhi Wang, Cong Yu, Simon Baumgartner, and Flip Korn. 2018. Relevant document discovery for fact-checking articles. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 525–533. International World Wide Web Conferences Steering Committee.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL HLT*, pages 1112–1122.

You Wu, Pankaj K. Agarwal, Chengkai Li, Jun Yang, and Cong Yu. 2014. Toward computational fact-checking. volume 7, pages 589–600. VLDB Endowment.

You Wu, Pankaj K. Agarwal, Chengkai Li, Jun Yang, and Cong Yu. 2017. Computational fact checking through query perturbations. *ACM Transactions on Database Systems (TODS)*, 42(1):4:1–4:41.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

Roberto Zanoli and Silvia Colombo. 2017. A transformation-driven approach for recognizing textual entailment. *Natural Language Engineering*, 23(4):507–534.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.