

知識グラフに基づくルールベースよる Fact Verification とその 拡張手法の考察

梶井裕貴¹ 滝口哲也¹ 有木康雄¹

¹ 神戸大学大学院システム情報学研究科

235x075x@stu.kobe-u.ac.jp {takigu,ariki}@kobe-u.ac.jp

概要

Fact Verification とは, claim に誤った情報が含まれていないかを判定するタスクであり, 判定に必要な情報を検索するステップと, 検索された情報に基づき claim の真偽を判定するステップで構成される. 本論文では, 最終判定の説明性を高めるため, 知識グラフを活用し, 最終判定をルールベースで行う方法を提案する. FACTKG データセットによる実験では, ルールベースの手法でモデルベースの手法と同等の結果が得られた. さらに, ルールベースの判定で課題となる, 複数の知識トリプルを比較して判定を行う方法について議論する.

1 はじめに

近年では誤った情報の拡散が進んでおり, Fact Verification の需要が高まっている. Fact Verification とは, 文法的に正しい claim に, 誤った情報が含まれていないかを, 検索された外部情報に基づいて判定するタスクである [1]. 外部情報にテキストを利用する手法 [2] や, 表を利用する手法 [3] は存在しているが, 知識グラフを用いた研究はあまり進んでいない [4]. 知識グラフを用いる場合, entity と呼ばれるノードと relation と呼ばれるエッジを合わせた知識トリプルにより, 解釈しやすい判定根拠を示すことができる. Fact Verification において, 判定根拠の明示は, 判定の信頼性を確保する上で重要になる.

Fact Verification の代表的なデータセットとして, FEVER [5] が存在している. FEVER は外部情報として Wikipedia を扱っている. FEVER に取り組んでいる多くのシステムは, 外部情報から情報を取得する過程, 取得した情報に基づいて claim の真偽判定を行う過程の両方にニューラルモデルを用いている [6]. これらのシステムは, 判定根拠と判定結果を示すことはできるが, 判定根拠から判定結果に至る

までの過程の透明性が欠ける. Fact Verification では説明性が重要であり, 最終判定をルールベースで行い, 透明性を高めることが望ましい.

構造化されたデータベースである知識グラフとルールベースの判定は相性がよく, 知識グラフを活用したルールベースの Fact Verification が行われている [7, 8]. しかしこれらの手法では, 知識グラフをマイニングすることでルールを獲得しており, 知識グラフの更新があった場合に対応が困難である.

本論文では, 知識グラフの entity や relation に依存しない判定ルールを事前作成し, 知識グラフの更新に対応できるルールベース判定を行う. このルールは, FACTKG [4] で定義された, 知識グラフの構造に合わせた claim の分類を参考に作成した. FACTKG データセット, OpenDialKG [9] の実験結果では, モデルベースの手法と同等の結果を達成した. 一方で, 知識グラフに基づくルールで Fact Verification を行う上で課題となる, 複数文に対する判定や複数の知識トリプルを比較する判定について議論を行う.

2 FACTKG

FACTKG (Fact Verification via Reasoning on Knowledge Graphs) [4] は, 外部情報に知識グラフを用いた Fact Verification のためのデータセットであり, claim を, 判定に必要な知識グラフの構造により 4 つの推論タイプ (One-Hop, Conjunction, Existence, Multi-Hop) に分類している (表 1). claim には, SUPPORTED か REFUTED のラベルが付与されている. ラベルが SUPPORTED である claim の一つの entity を別の entity へと置換することで, REFUTED が作成される. One-Hop については, claim 中の relation 置換が行われる場合もある. 各タイプに対し否定語 (not, no ...) が挿入され別タイプとして扱われることもあり, 計 8 つの推論タイプがある.

1. One-Hop: 知識トリプル一つで判定できるもの.

表 1 FACTKG における推論タイプ [4](r1: parentCompany, r2: Builder, r3: Operator, r4: location, m: Meyer Werft, s: AIDAstella, c: AIDA Cruises, p: Papenburg, n: New York)

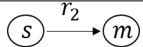
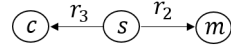
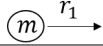


| Reasoning type | Claim example | Graph |
|------------------------|--|---|
| One-Hop | AIDAstella was built by Meyer Werft. |  |
| Conjunction | AIDA Cruise line operated the AIDAstella which was built by Meyer Werft. |  |
| Existence | Meyer Werft had a parent company. |  |
| Multi-Hop | AIDAstella was built by a company in Papenburg. |  |
| Conjunction (Negation) | AIDAstella was built by Meyer Werft, not in New York. |  |

表 1 の 2 行目では, (AIDAstella, builder, Meyer Werft) が存在すれば SUPPORTED となる.

- Conjunction: 複数の知識トリプルで判定できるもの. 3 行目では, (AIDAstella, Operator, AIDA Cruise), (AIDAstella, Builder, Meyer Werft) の 2 つが存在すれば SUPPORTED となる.
- Existence: claim 中の entity が特定の relation を持つかどうかで判定できるもの. 4 行目では, Meyer Werft が parentCompany を持てば SUPPORTED となる.
- Multi-Hop: Conjunction において, entity が一般名詞に変換されているもの. 5 行目では, (AIDAstella, Builder, x), (x, location, Papenburg) の両方の知識トリプルを満たす x が存在すれば SUPPORTED となる.

Negation は, これら 4 つの推論タイプの claim に対して, 否定語を挿入することで作成される.

3 提案手法

Type Predictor, Negation Predictor, Relation Predictor と人手により作成したルールで構成される.

3.1 Predictor の役割

すべての予測器で, BERT[10](bert-base-uncased) を用いる. Predictor の概略を図 1 に示す. Type Predictor, Negation Predictor は適用するルールを決定するために使用する. Type Predictor は, claim が表 1 の 2~5 行目のどの推論タイプに該当するかを分類問題として判定する. Negation Predictor は否定語が挿入されているかを判定する. Relation Predictor は, claim, [SEP], entity を結合して Input とし, 判定に必要な知識グラフ上の relation をマルチラベル問題として予測する. claim に含まれる entity 毎に Input を作成し, 予測された relation を全て合わせりレー

ションセット R とする.

3.2 最終判定ルール

- One-Hop(図 2(a)): claim 中の entity が知識グラフ上で直接結合するかを検証する. 図 2(a) では, AIDAstella と Meyer Werft が知識グラフ上で結合しているかを検証し, 実際に結合していた例を示している. 結合しない場合は, 誤った知識トリプルをもとに claim が生成されていると判定し, 予測を REFUTED とする. 図 2(a) のように, 結合する場合は, 結合した relation がリレーションセット R に含まれていたかを検証し, 含まれていれば予測を SUPPORTED とする. この判定により, claim が entity 同士の関係性を誤る場合にも対応できる.
- Conjunction(図 2(a)): One-Hop と同様に行う.
- Existence(図 2(b)): claim 中の entity が持つ relation を全て取得し, R と共通する relation が存在した場合, 予測を SUPPORTED とする.
- Multi-Hop(図 2(c)): R を並べ替えることで候補パスを生成する. claim 中の entity から候補パスで知識グラフの探索を行い, サブグラフを取得する. 図 2(c) の場合は, R が builder, location であり, 並び替えることで 2 個の候補パスが生成され, 知識グラフの探索を行っている. すべての entity がサブグラフに現れる場合, 予想を SUPPORTED とする. なお, 探索時のホップ数は 3 で固定する. これは, データセット FACTKG における最大のホップ数が 3 であり, さらに, 最終判定にニューラルモデルを用いないので, 入力可能トークンの上限を考慮する必要がないためである. ただし, 探索範囲が広すぎると, claim 中の entity が関係ない場合でも, 同じサブグラフに現れる確率が高くなり, 精度

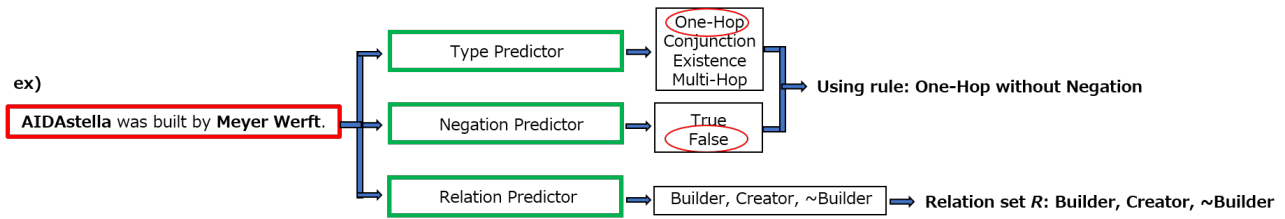


図 1 Type Predictor, Negation Predictor, Relation Predictor による適用ルール決定と relation 集合の決定例.

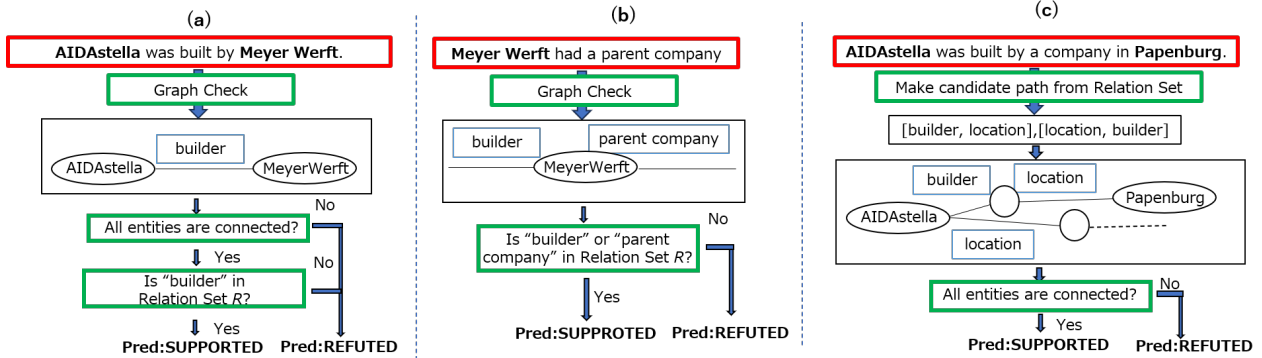


図 2 ルールベースによる Fact Verification 最終判定 (a)One-Hop,Conjunction(b)Existence(c)Multi-Hop

が低下するリスクがある。

仮定法を示すフレーズ (“I imagined that”, “I wish that”, “If only”) が含まれる場合は、予測ラベル SUPPORTED と REFUTED を反転させる。Negation 時、One-Hop と Existence では、図 2 の予測ラベルを反転させる。Conjunction と Multi-Hop のルールは補足資料に記載する。

4 実験

4.1 ベースライン

ベースライン (Model Based) は、FACTKG[4] のベースラインを用いる。知識グラフ探索は提案手法の Multi-Hop と同様に行うが、グラフ探索に必要なホップ数も予測する。取得されたサブグラフを用いて、ファインチューニングされた BERT(bert-base-cased)+線形層で最終判定 SUPPORTED/REFUTED を行う。

4.2 データセット

FACTKG[4] で作成されたデータセットと OpenDialKG[9] を使用する。

FACTKG の評価データには判定に必要な知識トリプルが付与されていないので、検証データを二分割し新たな検証データと評価データを作成した。OpenDialKG は、知識グラフに基づく対話データセットである。対話履歴と、それに対する応答、応答に用いられた知識トリプルが付与されている。

この応答を判定対象として用いる。本来の応答を SUPPROTED, 応答中の entity を別の entity へと置換したものを REFUTED とした。また、ベースラインの最終判定部分の学習と、全体の評価には、付与されている知識トリプル中の全ての entity が応答に現れているもののみを用いた。推論タイプとしては、One-Hop と Conjunction しか存在しないので、Type Predictor は用いない。否定語の挿入はされていないので、Negation Predictor も用いない。ベースラインにおけるホップ予測も正解を与える。

4.3 評価指標

予測ラベルだけにに基づく単純な正解率 (LA) に加え、FEVER Shared Task[11] にならい、Relation Predictor が誤る場合、情報検索が行えないとし、最終判定誤りとする Evidence Enhanced Label Accuracy(EELA) の評価を行う。ベースラインでは、ホップ予測時に予測したホップ数が、必要なホップ数よりも小さい場合にも判定誤りとして扱う。提案手法では、Type Predictor が誤った場合も判定誤りとして扱う。ただし、One-Hop を Conjunction か Multi-Hop, Conjunction を One-Hop か Multi-Hop と誤る場合は、別タイプのルールを適用可能なので、許容する。

4.4 実験結果

FACTKG における実験結果を表 2 に示す。提案手法は、ベースラインと同程度の精度になっている。

表 2 FactKG における Fact Verification 正解率. Nor は Negation なし, Neg はありの結果を示す.

| | | One-Hop | | Conjunction | | Existence | | Multi-Hop | |
|------------------|------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | | Nor | Neg | Nor | Neg | Nor | Neg | Nor | Neg |
| Model Based | LA | 0.9230 | 0.8932 | 0.9120 | 0.8641 | 0.7823 | 0.8007 | 0.7721 | 0.7678 |
| | EELA | 0.7423 | 0.7509 | 0.4703 | 0.5923 | 0.7823 | 0.8007 | 0.2999 | 0.3679 |
| Rule Based(Ours) | LA | 0.9301 | 0.9003 | 0.8842 | 0.8447 | 0.9440 | 0.9710 | 0.7165 | 0.7047 |
| | EELA | 0.7258 | 0.7616 | 0.4699 | 0.5194 | 0.9440 | 0.9710 | 0.3292 | 0.3109 |

これから、ルールベースの判定により、最終判定に至るまでのプロセスをより明確にしつつ、精度が維持できていることが分かる。

一方、Conjunction と Multi-Hop に Negation が含まれる場合には、ベースラインに比べて精度が低下している。誤った例を調査したところ、否定語の影響範囲を捉えられていない例もあったが、アノテーションが間違っている例も一定数確認できた。例えば、“The Accademia di Architettura di Mendrisio is not located in a country led by Johann Schneider-Ammann.” という claim については、(Accademia di Architettura di Mendrisio, country, Swizerland), (Johann Schneider Ammann, ~leaderName, Swizerland) の 2 つの知識トリプルが必要とされている。(Johann Schneider-Ammann, ~ leaderName) の edge entity は、Swizerland のみである。つまり、この claim のラベルは REFUTED であるべきだが、SUPPROTED とラベルが付与されている。この 2 つの推論タイプは、アノテーション方法自体が複雑であるため、データセットの検証が必要である。

OpenDialKG では、提案手法の方がいずれも良い結果となっている(表 3)。しかし、負例の作成方法がルールベースに有利な方法である点、実験に用いたデータ数が少ない点を考えると、提案手法がベースラインに対して、必ずしも有効な方法であるとは言えない。しかし、最終判定時の学習データが不足している場合でも、ルールベースであれば判定が可能であると言える。

5 ルールベースの課題と対応策

今回は FACTKG の推論タイプごとにルールを作成したが、複数の推論タイプが混じった claim は判定できない。そこで、claim から知識グラフを直接生成することが考えられる。データベースとの直接の比較が可能となり、推論タイプ判定も不要となる。ただし、否定語が含まれる場合は、直接の比較で判定が行えないので、今回のような推論タイプ別のルールで判定する必要がある。

また、ルールベースで判定する上で課題になるの

表 3 OpenDialKG における Fact Verification 正解率

| | | One-Hop | Conjunction |
|------------------|------|---------------|---------------|
| Model Based | LA | 0.8695 | 0.7660 |
| | EELA | 0.8243 | 0.6390 |
| Rule Based(Ours) | LA | 0.9901 | 0.9788 |
| | EELA | 0.9457 | 0.8404 |

は、知識グラフ上で比較が求められる判定である。例えば、“Tokyo Sky Tree is taller than Tokyo Tower” という claim を判定するには、(Tokyo Sky Tree, height, 634m), (Tokyo Tower, height, 333m) の二つの知識トリプルから推論を行って判定を行う必要がある。しかし、今回のルールでは、単に entity が知識グラフ上で結合しているか否かを確認しているので、この種の claim を判定することはできない。

解決策として、Link Prediction により relation を追加することが考えられる。(Tokyo Sky Tree, height, 634m), (Tokyo Tower, height, 333m) から、“taller” という relation を追加できると判定できれば、(Tokyo Sky Tree, taller, Tokyo Tower) という知識トリプルを作成可能になる。この知識トリプルを作成すると、one-hop のルールが適用可能になり、ルールベースで判定ができる。Link Prediction には、GPT-4.0 の利用を想定している。

6 おわりに

本論文では、知識グラフを活用し、ニューラルモデルを用いず事前作成したルールによって Fact Verification の最終判定を行う方法を提案した。FACTKG, OpenDialKG による実験では、モデルベースの方法と同等以上の結果を得られることが分かった。また、ルールベースの課題である、知識グラフ上で比較を伴うような問題を解決するための手法として、Link Prediction による解決を提案した。今後は、このような知識グラフ上の比較を伴う問題のデータセットを作成し、この手法が有効かを検証したい。ルール自体の強化として、構文解析を行うことも考えられる。構文解析により、entity 同士のつながりを捉えれば、推論タイプの判定に利用できたり、Multi-Hop 時のホップ数の確定が行える。

謝辞

本研究の一部は、JSPS 科研費 JP21H00906 の支援を受けたものである。

参考文献

- [1] Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. GEAR: Graph-based evidence aggregating and reasoning for fact verification. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 892–901, Florence, Italy, July 2019. Association for Computational Linguistics.
- [2] Jungsoo Park, Sewon Min, Jaewoo Kang, Luke Zettlemoyer, and Hannaneh Hajishirzi. FaVIQ: Fact verification from information seeking questions. In **ACL**, 2022.
- [3] Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In Rami Aly, Christos Christodoulopoulos, Oana Cocarascu, Zhijiang Guo, Arpit Mittal, Michael Schlichtkrull, James Thorne, and Andreas Vlachos, editors, **Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)**, pp. 1–13, Dominican Republic, November 2021. Association for Computational Linguistics.
- [4] Jiho Kim, Sungjin Park, Yeonsu Kwon, Yohan Jo, James Thorne, and Edward Choi. FactKG: Fact verification via reasoning on knowledge graphs. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 16190–16206, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [5] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
- [6] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Fine-grained fact verification with kernel graph attention network. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 7342–7351, Online, July 2020. Association for Computational Linguistics.
- [7] Naser Ahmadi, Joohyung Lee, Paolo Papotti, and Mohammed Saeed. Explainable fact checking with probabilistic answer set programming. **ArXiv**, Vol. abs/1906.09198, 2019.
- [8] Mohamed H. Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. Exfakt: A framework for explaining facts over knowledge graphs and text. **Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining**, 2019.
- [9] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 845–854, Florence, Italy, July 2019. Association for Computational Linguistics.
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In **North American Chapter of the Association for Computational Linguistics**, 2019.
- [11] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. The fact extraction and VERification (FEVER) shared task. In James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal, editors, **Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)**, pp. 1–9, Brussels, Belgium, November 2018. Association for Computational Linguistics.

A Conjunction 否定語挿入時のルール

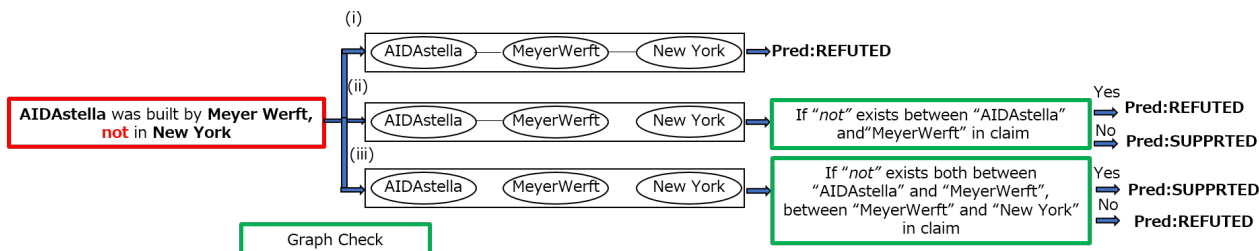


図3 ルールベースによる Fact Verification 最終判定で Conjunction に否定語が挿入された場合

FACTKG 上のデータセットに合わせ entity 数 3 の場合で説明を行う。Claim 中の entity が直接結合するか検証した場合、図 3 のように 3 つの場合分けがある。(i) の場合、すべて結合するにも関わらず、claim 中で否定していることになるので、予測を REFUTED とする。(ii) のように、一つの entity のみが結合しない場合は、否定語の位置によって予測が異なる。否定語が、Meyer Werft と New York の間にある場合は、誤った関係を否定していることになるので、予測を SUPPORTED とする。他の位置に否定語がある場合は、正しい関係を否定していることになるので、予測を REFUTED とする。(iii) の場合も同様に考え、すべての誤った関係に対して否定語が存在する場合に、予測を SUPPORTED とする。なお、否定語の影響範囲には、キーワードマッチングを用いて特定を行っている。

B Multi-Hop 否定語挿入時のルール

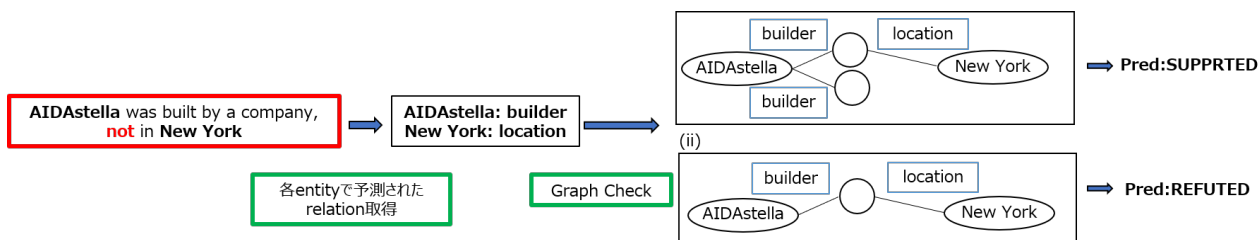


図4 ルールベースによる Fact Verification 最終判定で Multi-Hop に否定語が挿入された場合

entity 数 2, ホップ数 1 の場合で説明を行う。図 4 のように、リレーションセットを作成する前に、各 entity で、Relation Predictor が予測した最も確率の高い relation を取得する。もし最も確率の高い relation を、その entity が知識グラフ上で持たない場合は、次に確率が高い relation を選ぶ。その relation で知識グラフを探索し、edge entity を取得する。否定語が関係していない entity が持つ edge entity によって予測を変化させる。(i) のように、否定語が関係していない entity (“AIDAstella”) が、共通していない edge entity を持つ場合、予測を SUPPORTED とする。(ii) のように、共通するものしか edge entity を持たない場合は、予測を REFUTED とする。また、もし claim 中の entity の両方に否定語が影響する場合、予測を SUPPORTED とする。

EELA を計算する際には、各エンティティで選択した edge entity を探索するための relation が、正解の relation と一致するかを確認している。