# Détection d'affirmation et de contre-affirmation dans la littérature scientifique

Wenjing Deng

# Détection d'affirmation et de contre-affirmation dans la littérature scientifique

**Wenjing**
**DENG**

Sous la direction de Cyril LABBÉ

Laboratoire : Laboratoire d'Informatique de Grenoble

UFR LLASIC
Département Sciences du langage

---

Mémoire de master 2 mention Sciences du langage - 30 crédits

Parcours : Industries de la langue

Année universitaire 2020-2021

# Détection d'affirmation et de contre-affirmation dans la littérature scientifique

**Wenjing**
**DENG**

Sous la direction de Cyril LABBÉ

Laboratoire : Laboratoire d'Informatique de Grenoble

UFR LLASIC
Département Sciences du langage

---

Mémoire de master 2 mention Sciences du langage - 30 crédits

Parcours : Industries de la langue, orientation recherche

Année universitaire 2020-2021

# Remerciements

Je voudrais remercier mon encadrant Cyril Labbé pour son aide et son soutien tout au long de ce stage. Je tiens à remercier Claude Ponton pour avoir accepté d'être mon enseignant référent stage, et qui m'a conseillée lors de la rédaction de ce mémoire. Je tiens à remercier Olivier Kraif pour avoir accepté de faire partie de mon jury.

Je souhaite remercier mes parents pour leur confiance, leur soutien et leurs encouragements tout au long de ma vie et de mes études.

**DÉCLARATION ANTI-PLAGIAT**

1. Ce travail est le fruit d'un travail personnel et constitue un document original.
2. Je sais que prétendre être l'auteur d'un travail écrit par une autre personne est une pratique sévèrement sanctionnée par la loi.
3. Personne d'autre que moi n'a le droit de faire valoir ce travail, en totalité ou en partie, comme le sien.
4. Les propos repris mot à mot à d'autres auteurs figurent entre guillemets (citations).
5. Les écrits sur lesquels je m'appuie dans ce mémoire sont systématiquement référencés selon un système de renvoi bibliographique clair et précis.

PRENOM : …Wenjing……

NOM : ……DENG……

DATE : ……2.9.2021……

# Sommaire

# Chapitre 1. Introduction

De nos jours, nous sommes exposés à de nombreuses informations avec le développement des réseaux sociaux, mais il est difficile pour le public de savoir si elles sont vraies ou fausses. Parfois, nous rencontrons même des désaccords entre elles. C'est presque la même situation dans le domaine scientifique où il y a de plus en plus d'articles scientifiques publiées, mais les chercheurs ont du mal à se tenir au courant des nouvelles découvertes et recherches. Il est aussi fréquent de trouver des affirmations contradictoires dans les articles scientifiques ou sur les réseaux sociaux.

Voici deux affirmations extraites respectivement de deux articles scientifiques :

**Exemple 1.**

*"This suggests that TiO2 nano-aerosol did not gain access in quantifiable amount to the CNS either across the BBB or through axonal translocation from the nasal mucosa" (Disdier et al., 2017)*

**Exemple 2.**

*"TiO2 NPs could not only pass through the BBB but also disrupt the integrity of the BBB." (Song et al., 2015)*

La nanoparticule TiO2 peut-elle traverser la BBB ? Il est évident que nous ne pouvons pas trouver une réponse exacte à partir de ces deux affirmations. Actuellement, la plupart des outils ou sites web, comme PolitiFact et Snopes, sont dédiés au domaine politique ou au domaine du journalisme qui sert à vérifier les propos tenus par les hommes politiques ou détecter les fausses informations propagées dans les médias. Cependant, nous ne parvenons pas à trouver des outils accessible au public qui nous permet de vérifier les affirmations comme celles ci-dessus. Pour ce faire, nous avons besoin de concevoir un système ou de créer un outil permettant de identifier les accords et les désaccords dans la littérature scientifique, et de détecter les arguments qui soutiennent ou contredisent une affirmation controversée ou ambiguë.

Wadden et al. (2020) ont introduit la tâche de vérification d'affirmations scientifiques, qui est similaire à notre sujet. Cette tâche consiste à analyser une affirmation scientifique et à sélectionner des justifications dans un corpus annoté qui contient des articles scientifiques. Inspiré par

cette tâche, nous décidons de construire un système qui sert à vérifier l'affirmation scientifique et à trouver des arguments pour la justifier.

Ce travail est organisé de la manière suivante. Le chapitre 2 porte sur les approches et les jeux de données en rapport avec notre sujet. Nous présenterons trois tâches : la vérification des faits, le calcul de similarité sémantique et l'inférence en langage naturel. Ensuite, nous expliquerons notre problématique (chapitre 3)ainsi que les approches et les ressources que nous avons utilisées pour créer notre système (chapitre 4). Nous spécifierons notre système et interpréterons les résultats obtenus dans le chapitre 5. Enfin, nous essayerons de proposer des pistes d'améliorations dans le chapitre 6 et conclurons cette étude dans le chapitre 7.

# Chapitre 2. État de l'art

Dans ce chapitre, nous présentons un tour d'horizon des approches et des jeux de données existants qui nous aident à trouver une solution permettant de vérifier une affirmation et l'annoter. Nous construisons le système qui repose sur trois phases : la phase de sélection de résumés, la phase de sélection de phrases, et la phase de prédiction d'étiquettes. Nous nous concentrons sur trois tâches qui est en lien avec notre sujet : 1) la vérification de faits (section 2.1), qui nous donne une idée globale, 2) le calcul de similarité sémantique (section 2.2), qui peut être utile pour mesurer la ressemblance entre une affirmation et un argument, et 3) l'inférence en langage naturel (section 2.3), qui sert à déduire la relation logique entre deux phrases et à attribuer une étiquette.

## 2.1 Vérification de faits

La vérification de faits, ou le « fact-checking » en anglais, est une technique utilisée par les journalistes pour évaluer la véracité des propos ou des affirmations de personnalités publiques. La présence de site de fact-checking, tels que Snopes et PolitiFact, permet de limiter la propagation de rumeurs ou de fausses informations sur le Web.

Voici un exemple :

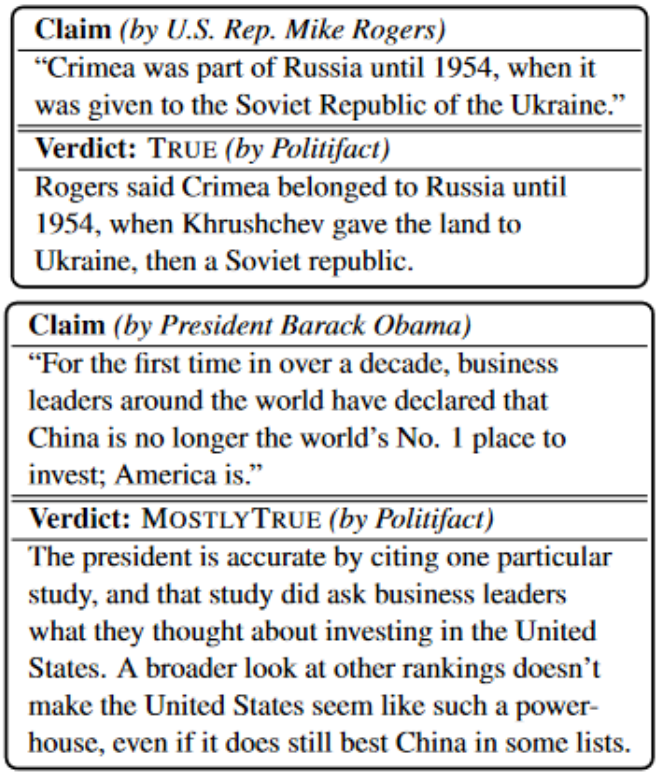


FIGURE 1: Exemple de fact-checking (Vlachos and Riedel, 2014*a*)

Aujourd'hui, la vérification de faits est en lien avec plusieurs domaines de recherche, y compris le traitement automatique du langage naturel, l'apprentissage machine, l'analyse sociologique, etc. Dans la pratique, elle est toujours traitée comme une tâche de classification qui permet de détecter la corrélation entre une phrase et un texte, et de déterminer si elles se contredisent ou si la phrase est étayée par le texte. Il existe déjà des tâches similaires comme la détection de positions (stance detection), la détection de fausses informations, etc.

### 2.1.1 Définition de Fact-checking

Selon Thorne and Vlachos (2018), trois éléments importants définissent cette tâche :

- l'entrée : ce qui doit être vérifiée

- la justification : ce qui est nécessaire pour la vérification

- la sortie : ce qui est attendue comme verdict

Concernant l'entrée, il y a généralement deux types d'entrée : le triplet et l'affirmation.

Le triplet est composé de trois éléments importants : le sujet, le prédicat et l'objet. Le triplet est souvent utilisé dans la tâche d'extraction de relation qui sert à représenter la relation séman-

tique entre deux entitées. Vlachos and Riedel (2015) ont extrait les relations entre mots et ont généré l'affirmation sous forme de triplet

Voici un exemple d'un triplet (Syed et al., 2018) :

<Albert_Einstein, received, Nobel_Prize_in_Physics>

Le deuxième type d'entrée est l'affirmation, qui est une phrase courte construite à partir de paragraphe ou de texte. L'affirmation peut être extraite directement de l'actualité, d'un article scientifique, d'un rapport ou même d'un engagement pris par des hommes politiques (Vo and Lee, 2020; Thorne et al., 2018). Elle peut également être généré par la réécriture d'articles (Wadden et al., 2020).

Voici un exemple de l'affirmation avec la justification :

| |
|---|
| **Claim 1**: Lopinavir / ritonavir have exhibited favorable clinical responses when used as a treatment for coronavirus. |
| **Supports**: . . . *Interestingly, after lopinavir/ritonavir (Kaletra, AbbVie) was administered, β-coronavirus viral loads significantly decreased and no or little coronavirus titers were observed.* |
| **Refutes**: *The focused drug repurposing of known approved drugs (such as lopinavir/ritonavir) has been reported failed for curing SARS-CoV-2 infected patients.* It is urgent to generate new chemical entities against this virus . . . |
| **Claim 2**: The coronavirus cannot thrive in warmer climates. |
| **Supports**: *...most outbreaks display a pattern of clustering in relatively cool and dry areas...This is because the environment can mediate human-to-human transmission of SARS-CoV-2, and unsuitable climates can cause the virus to destabilize quickly...* |
| **Refutes**: *...significant cases in the coming months are likely to occur in more humid (warmer) climates, irrespective of the climate-dependence of transmission and that summer temperatures will not substrantially limit pandemic growth.* |

FIGURE 2: Exemple de l'affirmation (Wadden et al., 2020)

Plusieurs types de justification sont utilisées dans les recherches précédentes. Rashkin et al. (2017) ont utilisé l'affirmation elle-même pour confirmer sa véracité et prend en compte la façon dont l'affirmation est écrite plutôt que les faits. Le graphe de connaissances contribue aussi à la vérification des faits. Nous pouvons extraire ou identifier des entités du graphe de connaissances. Vlachos and Riedel (2015) ont détecté le triplet sujet-prédicat-objet à partir des graphes de connaissances pour vérifier les chiffres dans les affirmations. Les textes, tels que les articles sur Wikipédia ou les revues scientifiques, peuvent également utilisés pour vérifier les affirmations. Par exemple, Ferreira and Vlachos (2016) ont utilisé les titres d'article pour savoir si un article est pour ou contre une affirmation.

La sortie peut être considérée comme une classification binaire qui attribue une étiquette marquée « Vrai » ou « Faux » comme dans la recherche de Nakashole and Mitchell (2014). Les étiquettes prédites dépendent de la relation logique entre l'affirmation et la justification, donc elles peuvent être divisées en trois catégories en fonction de différente relation. Par exemple,

dans Thorne et al. (2018), les chercheurs ont caractérisé les relations en trois classes globales :

- *SUPPORTS* : la justification soutient l'affirmation

- *REFUTES* : la justification contredise l'affirmation

- *NOT_ENOUGH_INFO* : la justification est ambiguë ou il n'y a aucun lien entre la justification (ou le texte original) et l'affirmation

### 2.1.2 Domaine du journalisme

Popat et al. (2018) ont proposé un système appelé DeClarE (Debunking Claims with Interpretable Evidence) qui sert à évaluer la crédibilité de la presse tout en fournissant des explications compréhensibles pour l'utilisateur. Quand il y a une affirmation à vérifier, le système peut rassembler sur Internet des articles pertinents et donner des preuves et des conclusions plausibles.

DeClarE est un système de bout-en-bout évalué sur 4 jeux de données tels que Snopes, PolitiFact, NewsTrust et SemEval-2017 Task 8 (RumourEval17). Il se compose principalement de deux parties. La première partie consiste à d'abord concaténer la moyenne des plongements de mots de l'affirmation et les plongements de mots de l'article, puis utiliser le mécanisme d'attention pour se concentrer sur les mots saillants permettant de confirmer l'affirmation et générer les poids d'attention de l'affirmation. La deuxième partie est d'utiliser un BiLSTM pour obtenir la représentation de l'article. Enfin, tous les éléments, y compris la représentation de l'article traitée avec le mécanisme d'attention, la représentation vectorielle de la source de l'affirmation ainsi que celle de la source de l'article, sont combinés en une seule représentation, afin de prédire la crédibilité de la presse et attribuer une étiquette classée en « Vrai » et « Faux ».

### 2.1.3 Domaine biomédical

La vérification scientifique (Scientific verification, SciVer) [1] est proposée par Wadden et al. (2020). Elle consiste à identifier d'abord des résumés pertinents pour une affirmation scientifique, puis à fouiller dans ces résumés pour sélectionner des phrases justificatives, enfin de prédire la relation entre l'affirmation et la phrase justificative. Un exemple est donné ci-dessous (voir figure 3).

Pour traiter cette tâche, les auteurs ont construit un jeu de données appelé SciFact. Il contient

1. https ://sdproc.org/2021/sharedtasks.htmlsciver

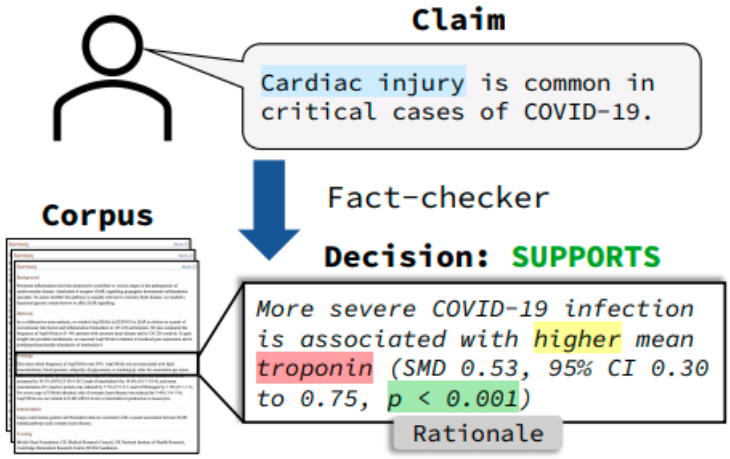


FIGURE 3: Exemple de Fact-checkeur (Pradeep et al., 2020)

des affirmations scientifiques annotées avec des phrase justificatives et des étiquettes, ainsi que des résumés qui peuvent soutenir ou contredire les affirmations. Nous allons présenter en détail SciFact dans la section 3.2.

Wadden et al. (2020) ont proposé le modèle de base VERISCI. Ce modèle utilise d'abord le TF-IDF, une méthode de pondération qui permet d'évaluer la pertinence d'un mot dans un document relativement à un corpus (Jones, 1972), afin de trouver les k résumés les plus pertinents. Ensuite, il entraîne le modèle sur le jeu de données SciFact pour détecter les phrases justificatives dans le résumé. Enfin, il entraîne le modèle sur deux jeux de données, FEVER et SciFact , pour classer la relation en trois catégories : « SUPPORT », « CONTRADICT », et « NOT_ENOUGH_INFO ».

Nous allons présenter quelques méthodes utilisées par des groupes de recherches évaluées dans la tâche SciVer.

Pradeep et al. (2020) ont proposé VERT5ERINI, un système en pipeline destiné à la vérification des affirmations scientifiques. Il exploite le modèle BM25 (Robertson et al., 2004) pour sélectionner 20 résumés candidats, dont les 3 résumés les plus pertinents sont identifiés par le modèle T5. Ensuite, il entraîne séparément le modèle T5 pour la sélection des phrases. Pour les étapes suivantes, il utilise également le modèle T5 sur le dataset SciFact mais le entraîne d'une manière différente. Voici une illustration qui décrit le pipeline de VERT5ERINI.

Li et al. (2020) ont présenté ParagraphJoint, un modèle qui traite cette tâche au niveau de document. Il génère les plongements de phrases en utilisant l'outil BioSentVec et récupère les 30 résumés les plus pertinents. Puis il entraîne le modèle RoBERTa-Large sur le dataset FEVER et le dataset SciFact dans le but de traiter la sélection des phrases et la prédiction des étiquettes

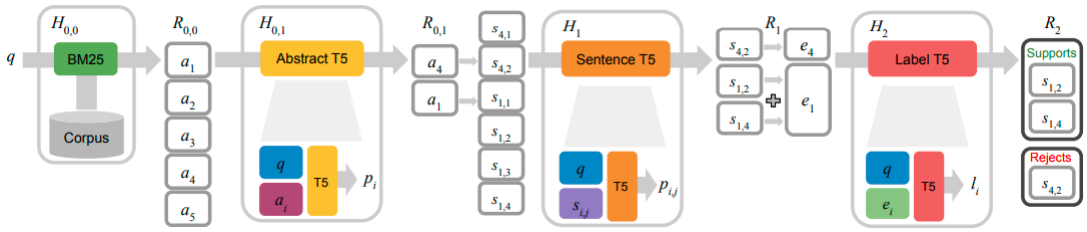


FIGURE 4: Illustration de système VERT5ERINI (Pradeep et al., 2020)

en même temps.

### 2.1.4 Domaine politique

Vlachos and Riedel (2014*b*) ont donné une définition précise de la vérification de faits dans le domaine politique. Cette tâche a pour but de confirmer la véracité des énoncés faits par des politiciens, des experts, etc.

Il existe deux approches de base pour traiter cette tâche. La première approche est de classer les déclarations en différentes catégories en utilisant des algorithmes qui apprennent à partir de corpus manuellement annoté.

Ensuite, inspirée par Agirre et al. (2013), une autre approche est de faire correspondre les énoncés à ceux qui ont déjà été vérifiés par des journalistes et d'attribuer l'étiquette en utilisant la méthode KNN (k plus proches voisins). Par contre, cette approche ne s'applique qu'aux énoncés existants et sert donc à détecter la répétition et la paraphrase des faux énoncés. Pour mieux repérer les énoncés faux, la solution possible est d'utiliser un corpus plus grand qui peut servir de source de tous les vrais énoncés (par exemple, Wikipédia).

### 2.1.5 Jeux de données de Fact-checking

Il existe plusieurs jeux de données liés à la vérification des faits. La plupart d'entre eux sont construits à partir de site de Fact-checking comme PolitiFact (Vlachos and Riedel, 2014*b*; Wang, 2017), Snopes (Popat et al., 2017) ou de Wikipédia comme FEVER (Thorne et al., 2018). Nous allons par la suite présenter cinq jeux de données populaires pour traiter cette tâche.

#### PolitiFact

PolitiFact [2] est un site web destiné à la vérification des faits. Il y a deux jeux de données

2. https ://www.politifact.com/

construits à partir du contenu de ce site : PolitiFact14 et PolitiFact17.

Créé par Vlachos and Riedel (2014*b*), PolitiFact14 est l'un des premiers jeux de données destiné à la vérification des faits dans le domaine du traitement automatique du langage naturel. Il contient 106 affirmations politiques extraites du site PolitiFact et du site Channel4 Fake News avec des documents de différents formats qui peuvent être utilisés comme preuves.

PolitiFact17, proposé par Wang (2017), est une extension de PolitiFact14. C'est un jeu de données à grande échelle qui contient 12800 affirmations confirmées avec des métadonnées correspondantes. Contrairement à PolitiFact14, la taille de Politi17 lui permet de contribuer à l'entraînement du système de vérification des faits.

**Snopes**

Comme PolitiFact, Snopes [3] est aussi un site web dédié à la vérification des faits. (Popat et al., 2017) ont construit un jeu de données à partir de Snopes. Ils ont extrait 4956 affirmations et fourni environ 30 documents pertinents pour chaque affirmation. Cependant, les documents peuvent contenir beaucoup d'informations inutiles puisqu'ils ne sont pas manuellement validés.

**FEVER**

Thorne et al. (2018) ont publié FEVER, un jeu de données à grande échelle, construit à partir de Wikipédia et destiné à l'extraction et à la vérification des faits. Il contient 185445 affirmations accompagné d'étiquettes annotées manuellement. Pour construire FEVER, les auteurs ont extrait des phrases comme preuves pour confirmer la véracité des affirmations et ont annoté les affirmations comme « SUPPORTED » ou « REFUTED ». S'il n'y a pas assez d'informations pour soutenir ou réfuter l'affirmation, elle est notée « NOT ENOUGH INFO ». Ci-dessous est un exemple extrait de FEVER.

**SemEval-2017 Task 8 (RumourEval17)**

RumourEval17 est proposé par Derczynski et al. (2017). Il comporte 297 rumeurs et 4519 messages (ou tweets) qui répondent à ces rumeurs pour créer un système permettant de prédire

3. https ://www.snopes.com/

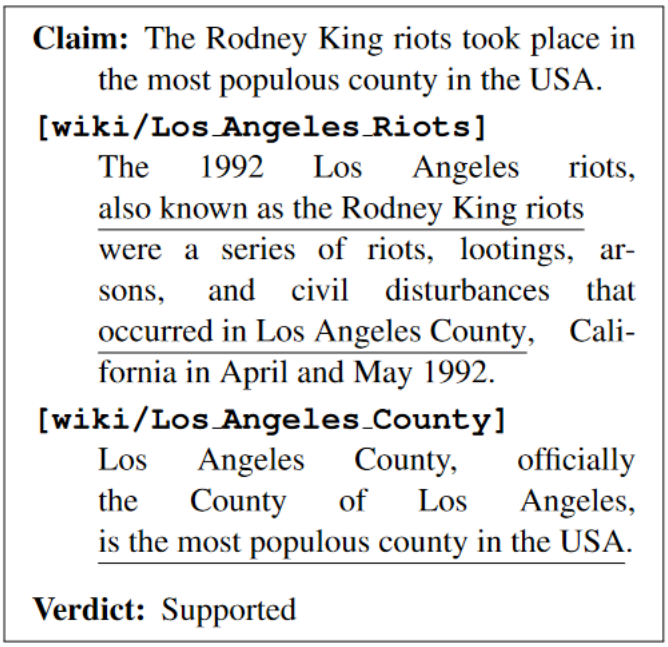


FIGURE 5: Exemple de FEVER (Thorne et al., 2018)

la véracité de la rumeur postée sur Twitter et attribuer une étiquette décrivant la véracité comme vraie ou fausse.

Voici un tableau qui regroupe les jeux de données les plus courants dans ce domaine :

| | Domaine | Affirmations | Documents | Citation |
|---|---|---|---|---|
| PolitiFact14 | Politique | 106 | - | (Vlachos and Riedel, 2014*b*) |
| PolitiFact17 | Politique | 12800 | - | (Wang, 2017) |
| Snopes17 | Domaine multiple | 4956 | 136085 | (Popat et al., 2017) |
| RumourEval17 | Domaine multiple | 297 | 4519 | (Derczynski et al., 2017) |
| FEVER | Domaine multiple | 185445 | 14533 | (Thorne et al., 2018) |
| SciFact | Biomédical | 1409 | 5138 | (Wadden et al., 2020) |

Tableau 1: Statistiques des jeux de données de Fact-checking

## 2.2 Calcul de similarité sémantique

La similarité sémantique est une mesure pour évaluer la ressemblance du point de vue de la signification entre des phrases. Les approches de similarité sémantique donnent souvent un score ou un classement de similarité, plutôt que de la traiter simplement comme un problème de classification binaire en indiquant si deux phrases sont similaires ou non Chandrasekaran and Mago (2021). Le score de similarité est généralement compris entre 0 et 1 : 0 signifie que les deux entrées ne sont pas du tout similaires, tandis que 1 signifie que elles sont identiques (Vlachos, 2010).

Dans l'application du traitement de langage naturel, la similarité sémantique est utilisée dans plusieurs domaines. Par exemple, pour le moteur de recherche comme Google, ou le forum informatique comme Quora, il faut évaluer la corrélation entre la requête proposée par l'utilisateur et la réponse candidate extraite du page web.

Pour calculer la similarité sémantique, le schéma le plus simple est d'abord d'encoder les phrases d'entrée et d'avoir leur représentations vectorielles, ensuite de calculer le score de similarité en utilisant des approches différentes. Nous allons par la suite discuter plusieurs approches utilisées pour le calcul de similarité sémantique.

### 2.2.1 Approches

#### TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) est une méthode de pondération qui permet d'évaluer l'importance d'un terme dans un document, relativement à une collection (Negre, 2013). Elle est aussi une méthode de base destinée à la recherche d'information et à la fouille de textes. Elle se compose de deux termes : TF et IDF. TF (Term Frequence) signifie la fréquence du terme dans le document. La TF est toujours normalisée puisque la fréquence du terme peut être plus élevée dans les documents longs. IDF est la fréquence inverse du document, qui signifie la rareté du terme, autrement dit, si le terme est moins fréquent au sein du corpus ou de l'ensemble de documents, il est plus discriminant. Le TF-IDF a donc tendance à exclure les mots courants et à conserver les mots importants. Voici la formule :

$$TF - IDF(t,d) = \frac{TF(t,d)}{DF(t)} = TF(t,d) \cdot IDF(t)$$

- $t$ : un terme ;

- $d$ : un document ;

- TF($t$,$d$) : la fréquence du terme dans le document ;

- DF($t$) : la fréquence du document ;

- IDF : la fréquence inverse du document.

**BM25**

BM25 est un algorithme amélioré à base de TF-IDF, permettant d'évaluer la corrélation entre la requête et le document. Dans le domaine de recherche d'information, le TF-IDF évalue l'importance d'un terme dans un document, tandis que le BM25 évalue la corrélation entre plusieurs termes et un document. Voici la formule :

$$BM25(D,Q) = \sum_{n}^{i=1} IDF(q_i) \cdot \frac{TF(q_i,D) \cdot (k_1+1)}{TF(q_i,D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgDL})}$$

- D : un document ;

- Q : une requête composée de plusieurs mots $q_1$, $q_2$, ..., $q_n$ ;

- $TF(q_i, D)$ : la fréquence de la requête dans le document ;

- $k_1$ et b : deux paramètres ;

- avgDL : la longueur moyenne du document.

**Similarité cosinus**

Supposons qu'il y a deux vecteurs X et Y, la similarité cosinus donne leur similarité en calculant le cosinus de leur angle dans un espace vectoriel. Plus le cosinus est proche de 1 et l'angle tend vers 0, plus les deux vecteurs sont similaires. La valeur de similarité cosinus est varié entre 0 et 1.

$$cosinus = \frac{X \cdot Y}{||X|| \cdot ||Y||}$$

**Distance de Manhattan**

Définie par Hermann Minkowski, la notion de distance de Manhattan vient de l'un des

arrondissements de la ville de New York, qui signifie la distance réelle entre deux points parcourue par un taxi quand il se déplace dans Manhattan. Elle est donc aussi appelé « distance de taxi ». Dans un espace vectoriel, nous suppons qu'il y a deux vecteurs $x = (x_1, x_2, ..., x_n)$ et $y = (y_1, y_2, ..., y_n)$, la distance de Manhattan est la somme des différences absolues de deux vecteurs, définie par :

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

**Distance euclidienne**

Dans un espace euclidien, la distance euclidienne est la distance la plus courte entre deux points. Dans un espace vectoriel, nous suppons qu'il y a deux vecteurs $x = (x_1, x_2, ..., x_n)$ et $y = (y_1, y_2, ..., y_n)$, la distance euclidienne est la racine carré de la somme des différences au carré entre deux vecteurs, définie par :

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

### 2.2.2 Jeux de données pour le calcul de similarité sémantique

**STS Benchmark**

Le jeu de données STS Benchmark (Semantic Textual Similarity benchmark) est proposé par Cer et al. (2017). Il est destiné à évaluer la similarité sémantique textuelle et est utilisé pour traiter plusieurs tâches comme la traduction automatique, la tâche de question-réponse, la recherche sémantique, le système de dialogue, etc. Il contient 4000 paires de phrases ainsi que leur score de similarité sémantique variant entre 1 et 5.

**SICK**

Le jeu de données SICK (Sentences Involving Compositional Knowledge) est proposé par Marelli et al. (2014). Il comporte en total 9840 paires de phrases annotées. Le score de similarité sémantique est varié entre 1 et 5.

## 2.3 Inférence en langage naturel

L'inférence en langage naturel (Natural language inference, NLI), parfois appelé la reconnaissance d'implication textuelle (Recognizing textual entailment, RTE), consiste à prédire la relation d'implication entre deux phrases : une prémisse et une hypothèse. Elle est souvent traitée comme une tache de classification multi-classes. Selon la sémantique textuelle, les relations d'implication sont souvent classées en trois classes : implication (entailment), contradiction, neutre (neutral). Supposons qu'une phrase prémisse soit marquée P et qu'une phrase hypothèse soit marquée H, les trois cas sont les suivants : P implique H, P contredit H, P et H ne sont pas liées.

### 2.3.1 Approches de NLI

Les approches de NLI sont généralement divisées en deux parties : les approches au niveau lexical, qui encodent d'abord les phrases d'entrée sous forme de vecteurs et analyse ensuite la relation entre vecteurs, et les approches au niveau de la phrase, qui considèrent principalement les relations de correspondance entre mots et ne reposent pas sur la représentation vectorielle sémantique de la phrase.

Nous allons tout d'abord travailler sur les approches au niveau lexical et puis les approches au niveau de la phrase.

#### Au niveau lexical

Selon les recherches précédentes, chaque terme contient différentes informations lexicales selon le contexte. Quand un terme est transformé en représentation vectorielle, il est possible que le système ne puisse pas reconnaître les différentes informations lexicales dans différents contextes.

Zhang et al. (2017) ont proposé le modèle CENN (Context-Enriched Neural Network) qui permet de reconnaître l'implication lexicale selon le contexte. Ce système utilise d'abord plusieurs vecteurs de mots provenant de différents contextes pour représenter les mots d'entrée. Ensuite, il intègre ces vecteurs de mots en utilisant différentes méthodes de combinaison et un mécanisme d'attention, et optimiser leurs poids pour prédire la relation d'implication entre les mots d'entrée. plusieurs jeux de données sont utilisés pour évaluer ce système, y compris Kotlerman2010 (Kotlerman et al., 2010), BLESS2011 (Baroni and Lenci, 2011), Baroni2012

(Androutsopoulos and Malakasiotis, 2010), Turney2014 (Turney and Mohammad, 2015) et Levy2014 (Levy et al., 2015).

**Au niveau de la phrase**

A part l'implication au niveau lexical, beaucoup de chercheurs travaillent sur l'implication au niveau de la phrase. Il existe deux méthodes : la méthode basé sur l'encodage de phrase et la méthode basée sur la mise en correspondance des mots.

Nous allons présenter tout d'abord la méthode basée sur l'encodage de phrase. Dans la recherche de Conneau et al. (2017), les auteurs ont présenté un schéma typique. Le schéma générique pour traiter la tâche NLI utilise souvent un encodeur de phrase partagé qui sert à transformer respectivement la prémisse et l'hypothèse en représentations vectorielles, marquée en $u$ et $v$. Ensuite, il extrait la relation entre les vecteurs de phrases $u$ et $v$ et génère un vecteur qui représente en même temps la prémisse et l'hypothèse en utilisant des méthodes différentes, comme la concaténation ($u$, $v$), le produit matriciel de Hadamard $u * v$ et la différence absolue $|u - v|$. Enfin, un classifieur utilise le vecteur généré à l'étape précédente pour prédire la relation d'implication entre les phrases d'entrée.

Liu et al. (2016) ont concentré sur le mécanisme d'attention. Leur approche est divisée en deux étapes. La première étape consiste à appliquer le mean-pooling, qui fait la moyenne des valeurs, sur le BiLSTM pour générer la première version de représentations de phrases. Ensuite, le mécanisme d'attention est employé sur les représentations générées à l'étape précédente afin d'attribuer un poids d'attentions à chaque mot dans la phrase. Enfin, un classifieur de SoftMax est utilisé pour décider si les phrases sont en « contradiction », « implication » et « neutre ». Les auteurs ont mené les expériences sur le jeu de données SNLI, qui démontre l'efficacité du mécanisme d'attention sur la tâche NLI.

Chen et al. (2016) ont présenté un modèle amélioré spécialement pour la tâche NLI en combinant BiLSTM et Tree-LSTM. Le modèle d'abord utilise le BiLSTM pour encoder les phrases d'entrée (la prémisse et l'hypothèse) et les transforment en représentations contextuelles. Ensuite, il exploite les informations d'inférence locale permettant de construire la prédiction finale en intégrant aussi des informations d'analyse syntaxique pour améliorer la performance. Le modèle est appris et évalué sur le jeu de données SNLI.

Concernant la méthode basée sur la mise en correspondance des mots, beaucoup de chercheurs s'intéressent au mécanisme d'attention pour transformer le problème de la relation entre

deux phrases en celui de la relation entre les mots correspondants des deux phrases.

Rocktäschel et al. (2015) ont présenté un nouveau modèle. Au lieu de travailler sur les représentations vectorielles de phrase, ils se sont concentrés sur les modèles de neurones qui peuvent ire les deux phrases d'entrée pour déterminer l'implication et ainsi reconnaître la relation d'implication entre paires de mots. Ils ont proposé un système qui s'appuie sur l'architecture LSTM et le mécanisme d'attention du mot à mot. Dans ce modèle, deux LSTMs sont utilisés pour encoder respectivement la prémisse et l'hypothèse. Inspiré par les recherches précédentes (Bahdanau et al. (2014), Hermann et al. (2015) et Rush et al. (2015)), les auteurs ont appliqué un mécanisme d'attention sur chaque mot de la phrase pour générer la représentation de paire de phrases. Cette méthode permet de renforcer la capacité du modèle à raisonner sur la relation entre des paires de mots ou de phrases sans avoir ses représentations vectorielles. Le modèle est évalué sur le jeu de données SNLI et plus performant que les autres modèles à base de LSTM.

### 2.3.2 Jeux de données de NLI

#### SNLI

SNLI (Stanford Natural Language Inference), proposé par Bowman et al. (2015), est le premier jeu de données annotées manuellement à grande échelle pour traiter la tâche de NLI. Il comporte 570000 paires de phrases annotées en fonction de leur relation logique (implication, contradiction, neutre). Voici un exemple extrait de SNLI (tableau 2) :

| Prémisse | Étiquette | Hypothèse |
|---|---|---|
| A soccer game with multiple males playing. | entailment | Some men are playing a sport. |
| An older and younger man smiling. | neutral | Two men are smiling and laughing at the cats playing on the floor. |
| A man inspects the uniform of a figure in some East Asian country. | contradiction | The man is sleeping. |

Tableau 2: Phrases d'exemple extraites de SNLI

**MNLI**

Multi-NLI (Multi-Genre Natural Language Inference), proposé par la tâche RepEval 2017 (Nangia et al., 2017), est une extension de SNLI. Il comprend 433000 paires de phrases couvrant des textes oraux et écrits, et permet d'évaluer les différents types de phrases comme présentés dans le tableau 3.

| Type | Prémisse | Étiquette | Hypothèse |
|---|---|---|---|
| Rapport | At the other end of Pennsylvania Avenue, people began to line up for a White House tour. | entailment | People formed a line at the end of Pennsylvania Avenue. |
| Lettre | Your gift is appreciated by each and every student who will benefit from your generosity. | neutre | Hundreds of students will benefit from your generosity. |
| Parole téléphonique | yes now you know if if everybody like in August when everybody's on vacation or something we can dress a little more casual or | contradiction | August is a black out month for vacations in the company. |

Tableau 3: Phrases d'exemple extraites de MNLI

**XNLI**

XNLI (Cross-lingual Natural Language Inference) est un jeu de données proposé par Conneau et al. (2018). Il utilise 7500 paires de phrases extraites de MNLI et les traduit en 15 langues différentes comme le français, l'espagnol, l'allemand, etc. L'objectif est d'exploiter des données d'entraînement en anglais pour faire des prédictions sur des données dans une autre langue (voir tableau 4).

| Langue | Prémisse | Étiquette | Hypothèse |
|---|---|---|---|
| Anglais | You don't have to stay there | entailment | You can leave |
| Français | La figure 4 montre la courbe d'offre des services de partage de travaux | entailment | Les services de partage de travaux ont une offre variable. |

Tableau 4: Phrases d'exemple extraites de XNLI

Dans ce chapitre, nous avons fourni un bref état de l'art des approches existantes liées à notre problématique. Nous avons également présenté les jeux de données les plus utilisés en pratique, afin de trouver des solutions permettant de confirmer la véracité de l'affirmation dans la littérature scientifique. Nous allons par la suite décrire la problématique ainsi que le jeu de données que nous avons utilisé dans notre étude.

# Chapitre 3. Contexte et problématique

Dans ce chapitre, nous allons présenter notre problématique et analyser le jeu de données utilisé dans cette recherche.

## 3.1 Présentation de la tâche

Notre problématique consiste à trouver une solution pour détecter les affirmations et les contre-affirmations dans la littérature scientifique. Supposons qu'il y a une affirmation, la tâche est de trouver des phrases à partir d'un corpus qui nous aide à confirmer sa véracité et reconnaître la relation d'implication entre elles.

| | |
|---|---|
| **Affirmation** | 1/2000 in UK have abnormal PrP positivity. |
| **Corpus** | "... SAMPLE 32,441 archived appendix samples fixed in formalin and embedded in paraffin and tested for the presence of abnormal prion protein (PrP). **RESULTS Of the 32,441 appendix samples 16 were positive for abnormal PrP, indicating an overall prevalence of 493 per million population (95% confidence interval 282 to 801 per million)**... " |
| **Relation** | **SUPPORT** |
| **Affirmation** | APOE4 expression in iPSC-derived neurons increases AlphaBeta production and tau phosphorylation, delaying GABA neuron degeneration. |
| **Corpus** | "... Efforts to develop drugs for Alzheimer's disease (AD) have shown promise in animal studies, only to fail in human trials, suggesting a pressing need to study AD in human model systems. **Using human neurons derived from induced pluripotent stem cells that expressed apolipoprotein E4 (ApoE4), a variant of the APOE gene product and the major genetic risk factor for AD, we demonstrated that ApoE4-expressing neurons had higher levels of tau phosphorylation, unrelated to their increased production of amyloid-03b2 (A03b2) peptides, and that they displayed GABAergic neuron degeneration**... " |
| **Relation** | **CONTRADICT** |

Tableau 5: Exemples extraits de SciFact (Wadden et al., 2020)

## 3.2 Jeu de données

Le jeu de données SciFact est proposé par Wadden et al. (2020) pour la tâche SciVer [4]. Il contient 1409 affirmations scientifiques et un corpus de 5183 résumés provenant d'articles vérifiés par les chercheurs dans le domaine biomédical. Chaque affirmation est justifiée par des phrases extraites de corpus.

**Affirmations**

Les affirmations sont divisées en trois sous-ensembles :

- un ensemble de données d'apprentissage
- un ensemble de données de développement
- un ensemble de données de test

Les données d'apprentissage et les données de développement contiennent des phrases justificatifs et des étiquettes indiquant la relation d'implication entre l'affirmation et la phrase justificatif. Malheureusement, bien que le défi soit relevé, les données d'essai étiquetées ne sont toujours pas accessibles au public. Les étiquettes sont divisés en trois classes : « SUPPORT », « NEI » (Not Enough Info) et « CONTRADICT ». Un exemple de schéma de l'affirmation est donné dans le listing 1.

Chaque affirmation se compose de quatre éléments principaux : un identifiant (*id*), une phrase affirmation (*claim*), un ensemble de justifications (*evidence*) et une liste des identifiants de résumés (*cited_doc_ids*). Chaque justification contient une ou plusieurs collections de phrases justificatives représentées par leur indices dans le résumé, une étiquette indiquant la relation d'implication et l'identifiant de résumé. Dans le listing 1, chaque collection de justification correspond à un résumé. Dans le résumé 11328820, l'ensemble de phrases 7 et 9 contredit l'affirmation 263. Dans le résumé 30041340, l'ensemble de phrases 0 et 1 réfute l'affirmation 263 tandis que la phrase 11 la réfute à elle seule.

4. https ://sdproc.org/2021/sharedtasks.htmlsciver

```
{
    "id": 1,
    "claim" : "0-dimensional biomaterials show inductive
     ↪  properties.",
    "evidence" : {},
    // le dictionnaire vide représente l'étiquette NEI (Not Enough
     ↪  Info)
    "cited_doc_ids": [31715818]
}
{
    "id": 263,
    "claim": "Citrullinated proteins externalized in neutrophil
     ↪  extracellular traps act indirectly to disrupt the
     ↪  inflammatory cycle.",
    "evidence": {
        "11328820": [ // identifiant de résumé
            { "sentences": [7, 9], // indices de phrase
            "label": "CONTRADICT" }
        ],
        "30041340": [
            { "sentences": [0, 1],
            // l'ensemble de phrases 0 et 1 réfute l'affirmation
             ↪  263
            "label": "CONTRADICT" },
            { "sentences": [11],
            "label": "CONTRADICT" }
        ]
    },
  "cited_doc_ids": [
        11328820,
        30041340,
        14853989
    ]
}
```

Listing 1: Exemple d'une affirmation avec des phrases justificatives extraites de corpus

Comme présenté dans la figure 6, l'affirmation est générée par la réécriture de résumés. Chaque document dans la liste de résumés cités (*cited_doc_ids*) peut contenir des phrases utilisées pour générer l'affirmation, mais il est possible que les annotateurs ne trouvent pas de phrases justificatives dans ces résumés. Dans la figure 1, par exemple, l'affirmation 263 est générée à partir de trois résumés (11328820, 30041340 et 14853989), mais les annotateurs ont trouvé des phrases justificatives dans les résumés 11328820 et 30041340 mais pas dans 14853989.

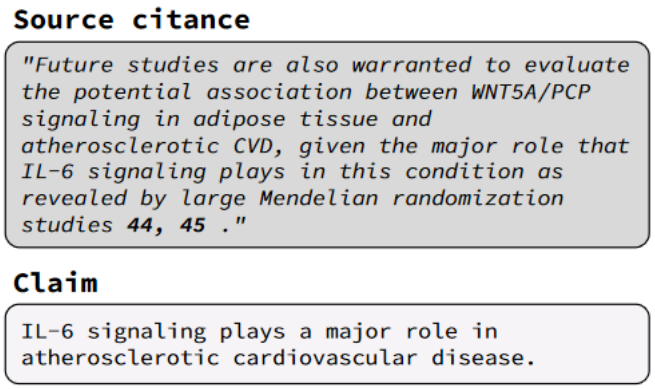


FIGURE 6: Exemple d'une affirmation générée à partir d'un résumé (Wadden et al., 2020)

Chaque affirmation n'est liée qu'à une seule classe, soit « CONTRADICT », soit « SUPPORT ». Selon le tableau 6, il y a un total de 1409 affirmations, dont 809 pour entraîner le modèle, 300 pour améliorer le modèle et 300 pour tester le modèle. La distribution des classes est déséquilibrée dans les données d'apprentissage et les données de développement, où les classes « SUPPORT » et « NEI » sont plus nombreuses que la classe « CONTRADICT ».

| | SUPPORT | NEI | CONTRADICT | Total |
|---|---|---|---|---|
| Données d'apprentissage (Train set) | 332 | 304 | 173 | 809 |
| Données de développement (Dev set) | 124 | 112 | 64 | 300 |
| Données de test (Test set) | 100 | 100 | 100 | 300 |
| All | 556 | 516 | 337 | 1409 |

Tableau 6: Distribution des étiquettes dans les jeux de données d'apprentissage, de développement et de test

Nous calculons la distribution des indices de phrases justificatives. Selon l'histogramme dans la figure 7, la majorité des phrases justificatives se trouve au début du résumé, surtout dans les dix premières phrases.

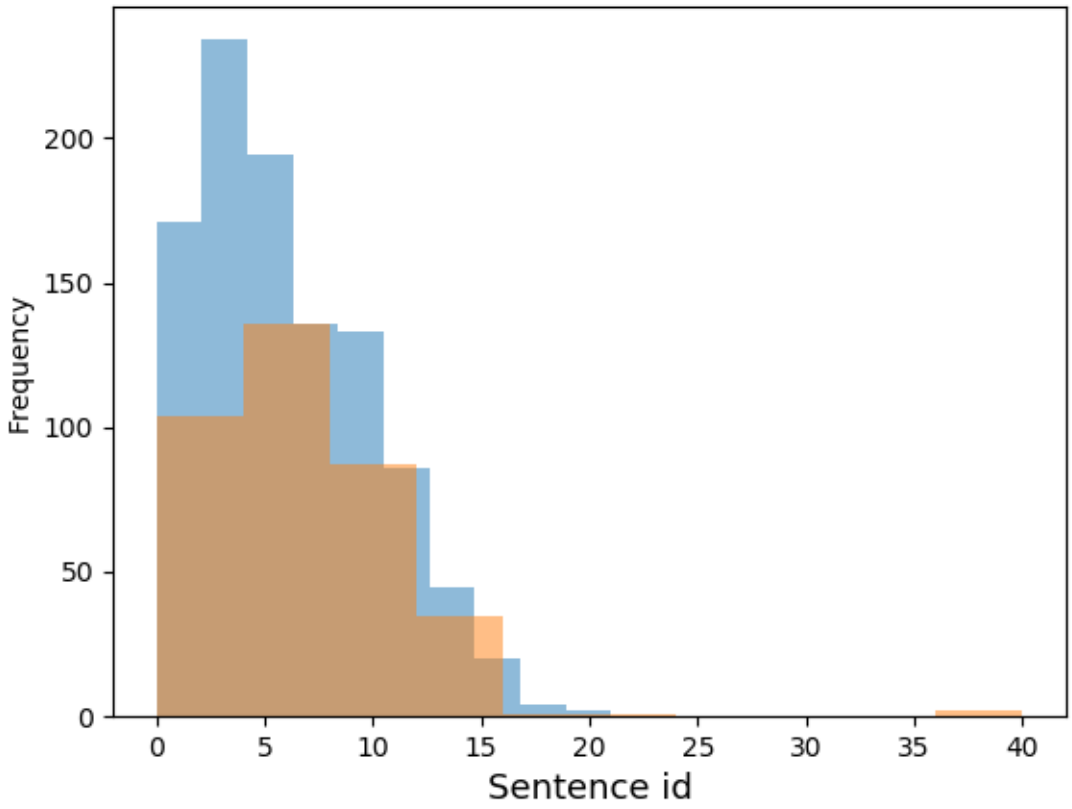


FIGURE 7: Histogramme de distribution des indices de phrases justificatives

La partie bleue représente l'ensemble de données d'apprentissage et la partie orange représente l'ensemble de données de développement.

Nous analysons aussi les données d'apprentissage et les données de développement. Dans les tableaux 7 et 8, nous pouvons constater que la plupart des affirmations sont générées à partir d'un seul résumé et que plus de la moitié des affirmations sont justifiées par un résumé. 46,9% et 53,1% des affirmations sont justifiées par une phrase. Plus de 90% des résumés justificatifs contiennent une phrase justificative.

| | 0 | 1 | 2 | 3+ | Moyenne |
|---|---|---|---|---|---|
| Nombre de résumés cités par affirmation | — | 732 (90.4%) | 53 (6.5%) | 24 (2.9%) | 1.13 |
| Nombre de résumés justificatifs par affirmation | 304 (37.5%) | 468 (57.8%) | 20 (2.4%) | 17 (2.1%) | 1.11 |
| Nombre de phrases justificatives par affirmation | — | 265 (46.9%) | 182 (32.2%) | 117 (20.7%) | 1.81 |
| Nombre de phrases justificatives par résumé | — | 897 (93.7%) | 53 (5.5%) | 7 (0.7%) | 1.07 |

Tableau 7: Statistiques des données d'apprentissage de SciFact

| | 0 | 1 | 2 | 3+ | Average |
|---|---|---|---|---|---|
| Nombre de résumés cités par affirmation | — | 276 (92%) | 15 (5%) | 9 (3%) | 1.13 |
| Nombre de résumés justificatifs par affirmation | 112 (37.3%) | 178 (59.3%) | 4 (1.3%) | 6 (2%) | 1.11 |
| Nombre de phrases justificatives par affirmation | — | 111 (53.1%) | 57 (27.2%) | 41 (19.6%) | 1.75 |
| Nombre de phrases justificatives par résumé | — | 313 (92.6%) | 22 (6.5%) | 3 (0.8%) | 1.08 |

Tableau 8: Statistiques des données de développement de SciFact

**Corpus (Résumés)**

Le corpus contient 5138 résumés provenant d'articles scientifiques dans le domaine biomédical et vérifiés par les chercheurs. Chaque document se compose de quatre éléments : un identifiant de document (*doc_id*), le titre du document (*title*) et le résumé du document (*abstract*).

Les statistiques du corpus sont présentées dans le tableau 9. Le résumé le plus court contient 3 phrases alors que le plus long contient 367 phrases. Plus de 70% des résumés ne comportent pas plus de 10 phrases. Il n'y a que 36 résumés qui contiennent plus de 21 phrases, dont 5 contiennent plus de 50 phrases.

```
{
    "doc_id": 4983,
    "title": "Microstructural development of human newborn cerebral
     ↪  white matter assessed in vivo diffusion tensor magnetic
     ↪  resonance imaging.",
    "abstract": [
        "Alterations of the architecture of cerebral white matter in
         ↪  the developing human brain can affect cortical
         ↪  development and result in functional disabilities.",
        ...
        ]
}
```

Listing 2: Extrait du corpus

| | Total | Min | Max | Moyenne |
|---|---|---|---|---|
| | 5183 | 3 | 367 | 8 |
| Nombre de phrases par résumé | 1-10 | 11-20 | 21-50 | 51+ |
| | 3854 (74.3%) | 1293 (24.9%) | 31 (0.5%) | 5 (0.09%) |

Tableau 9: Statistiques du corpus SciFact

Il y a 5183 résumés dans le corpus, dont 5147 (99,2%) ne contiennent pas plus de 20 phrases. Le résumé le plus long contient 367 phrases, tandis que le plus court ne contient que 3 phrases.

## 3.3 Évaluation

Nous évaluons les résultats à deux niveaux de granularité : l'évaluation du niveau de phrase et l'évaluation du niveau de résumé. Elles sont définies par Wadden et al. (2020). Nous allons expliquer les métriques à l'aide d'un exemple.

Nous avons une affirmation annotée et une prédiction dans les listings 3 et 4. L'affirmation annotée (voir le listing 3) contient quatre phrases justificatives provenant de deux résumés, dont les phrases 5 et 6 constituent un tout et soutiennent conjointement cette affirmation. L'affirmation prédite (voir le listing 4) contient aussi quatre phrases justificatives provenant de deux résumés, dont les phrases 5, 10 et 12 soutiennent conjointement cette affirmation.

**Évaluation du niveau de résumé**

L'objectif de l'évaluation du niveau de résumé est de savoir si le système peut reconnaître les résumés qui soutiennent ou réfutent l'affirmation. Pour évaluer si une phrase justificative est correctement annotée, il faut remplir trois conditions : (1) cette phrase est sémantiquement liée à l'affirmation, (2) son étiquette prédite est identique à celle manuellement annotée dans le jeu de données, et (3) au moins un ensemble de phrases justificatives est identifié

Dans le listing 4, le résumé 66 est correctement prédit car son étiquette prédite correspond à celle de l'annotation et il contient au moins une collection de phrases justificatives (collection [10]). Cependant, le résumé 99 est une fausse prédiction puisqu'il ne fait pas partie de la collection de résumés justificatifs (66 et 88).

**Évaluation du niveau de phrase**

L'évaluation du niveau de phrase vise à évaluer la capacité du système à correctement et exactement identifier les phrases justificatives à partir du corpus. Il y a aussi trois conditions à satisfaire : (1) la phrase provient d'un résumé lié à l'affirmation, (2) son étiquette prédite est identique à celle manuellement annotée dans le jeu de données, et (3) toutes les phrases dans le même ensemble sont également identifiées.

Dans le résumé 66, la phrase 10 est correct car elle fait partie de la collection [10] dans le listing 3, et la prédiction comprend la collection entière [10]. Par contre, les phrases 5 et 12 sont incorrectes. La phrase 5 appartient à la collection [5, 6] mais le système a omis la phrase 6 dans

la prédiction, cette prédiction est donc fausse. La phrase 12 est incorrecte parce qu'elle ne fait pas partie des phrases justificatives. Dans le résumé 99, la phrase 2 est incorrecte puisque le résumé 99 ne fait pas partie des résumés justificatifs.


```
{   "id": 261,
    "claim": "Chronic aerobic exercise alters endothelial
    function",
    "evidence": {
        "66": [
            {"sentences": [5, 6],
            "label": "SUPPORT"},
            {"sentences": [10],
            "label": "SUPPORT"}
        ],
        "88": [
            {"sentences": [12],
            "label": "SUPPORT"}
        ]
    },
    "cited_doc_ids": [66, 88]
}
```


Listing 3: Affirmation annotée


```
{   "id": 261,
    "evidence": {
        "66": [
            {"sentences": [5, 10, 12],
            "label": "SUPPORT"}],
        "99": [
            {"sentences": [2],
            "label": "REFUTES"}]
        }
}
```


Listing 4: Prédiction

# Chapitre 4. Architectures et modèles

## 4.1 Architectures

Dans ce chapitre, nous allons présenter les approches et les modèles que nous avons choisis pour la vérification des affirmations dans la littérature scientifique.

Un schéma de système est donné en ci-dessous pour donner une idée générale sur les procédures principales (voir figure 8) .

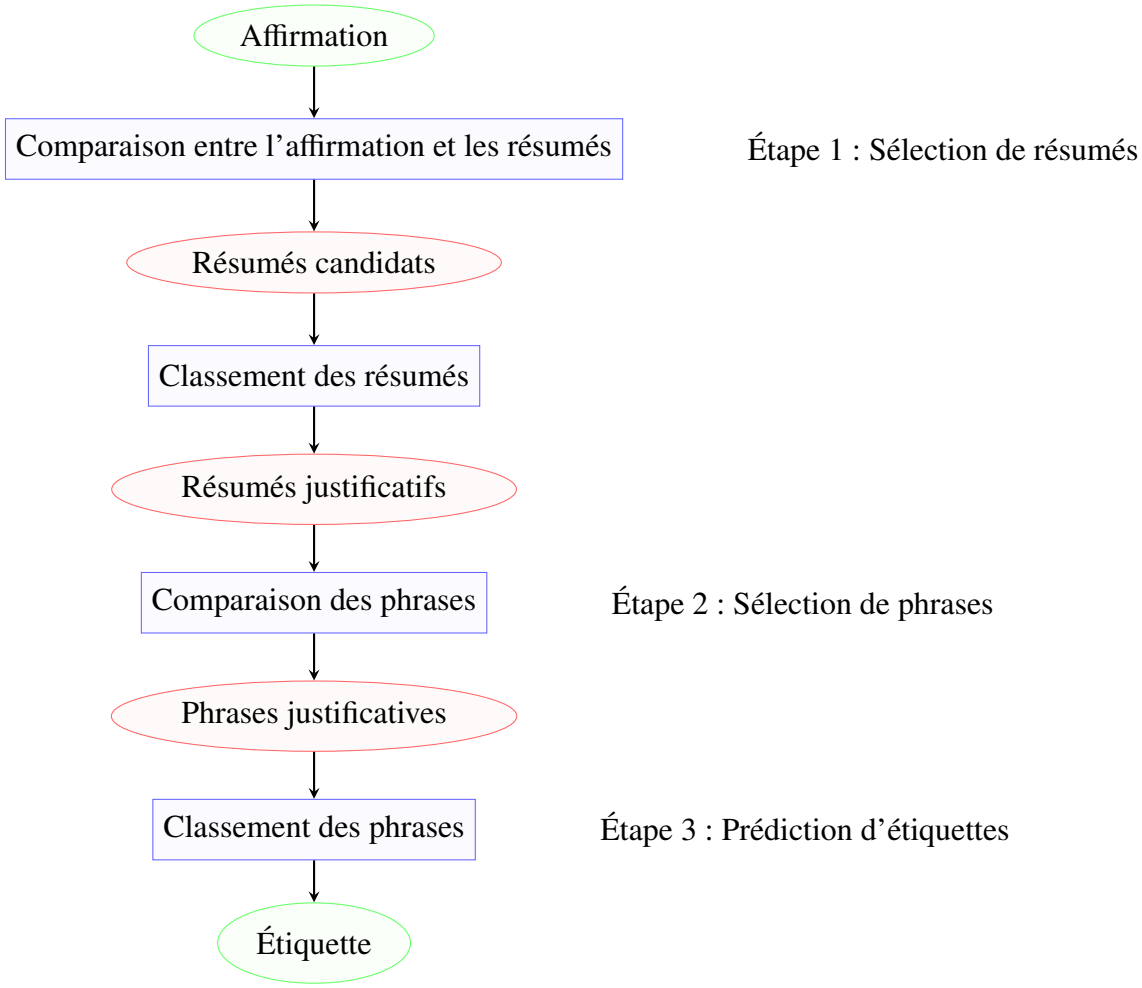


FIGURE 8: Procédures principales pour la vérification des affirmations dans la littérature scientifique

Notre système se décompose en trois étapes distinctes :

- la sélection de résumés
- la sélection de phrases
- la prédiction d'étiquettes

Dans un premier temps, notre système prend une affirmation et un corpus (un résumé de l'article scientifique) en entrée. Elles sont encodées par un bi-encodeur et transformées en représentations vectorielles. Ici, les vecteurs générés sont utilisés pour sélectionner des résumés candidats, qui est par la suite analysé par un cross-encodeur pour trouver les résumés pertinents (résumés justificatifs). Ensuite, chaque phrase dans les résumés sélectionnés est encodés par un autre bi-encodeur pour reconnaître des phrases qui peuvent soutenir ou réfuter la phrase d'entrée (phrases justificative). Enfin, un cross-encodeur est utilisé pour classer la paire de phrases (l'affirmation et la phrase justificative) en trois catégorie : « SUPPORTED », « REFUTED », et « NO_INFO ».

Nous utilisons Sentence Transformers (Reimers and Gurevych, 2019), un framework de Python destiné à la génération des représentations vectorielles de phrases ou de textes. Nous utilisons également l'architecture Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) pour construire notre système. Cette architecture s'appuie sur le réseau de neurones siamois : deux modèles BERT sont utilisés pour encoder respectivement deux phrases d'entrée pour obtenir la représentation vectorielle de chaque phrase. Les deux modèles BERT sont considérés comme un modèle puisqu'ils partage les mêmes configurations. Le résultat produit peut être utilisé pour le calcul de similarité sémantique ou la tâche de classification.

Pour la sélection de résumés et de phrases, nous utilisons l'architecture SBERT avec la fonction objectif de régression. Selon Figure 9, nous utilisons tout d'abord le modèle BERT pour générer les représentations vectorielles de phrases de deux phrases d'entrée A et B. Après l'encodage de phrases, l'opération de pooling est utilisé pour obtenir un vecteur fixe. Enfin, une similarité sémantique est calculée en utilisant le cosinus et un score de similarité est retourné.

Pour la prédiction d'étiquettes, nous recourons à l'architecture SBERT avec la fonction objectif de classification. Comme présenté dans la figure 10, la phase d'encodage de phrases est la même avec l'architecture présentée ci-dessus. La seule différence est qu'un classifieur Softmax analyse les représentations générées et classe les phrases en différentes catégories.

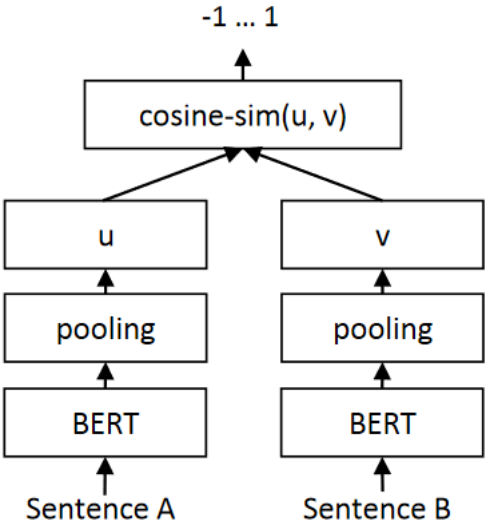


FIGURE 9: Architecture SBERT pour le calcul de similarité sémantique (Reimers and Gurevych, 2019)

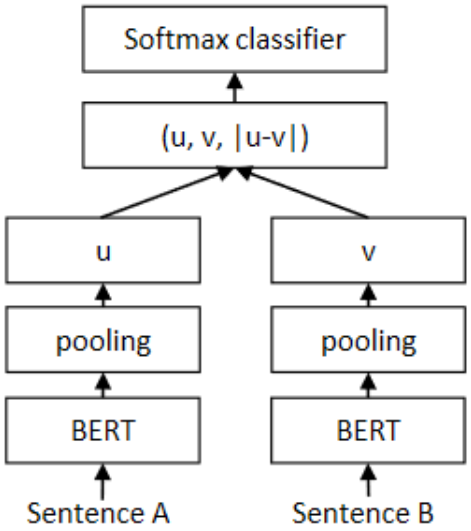


FIGURE 10: Architecture SBERT pour la classification (Reimers and Gurevych, 2019)

Contrairement à l'architecture de modèle BERT traditionnel dans la figure 11, le modèle BERT prend deux phrases en entrée et les concatène pour calculer la similarité sémantique textuelle. Le SBERT simplifie la phase de préparation avant le calcul. Avec l'architecture SBERT, les représentations vectorielles générées peuvent être stockées dans un dictionnaire. Pour calculer la similarité entre phrases, il suffit d'utiliser directement les représentations stockées au lieu d'encoder les phrases à chaque fois avant le calcul. L'avantage important est donc la réduction du temps de calcul.

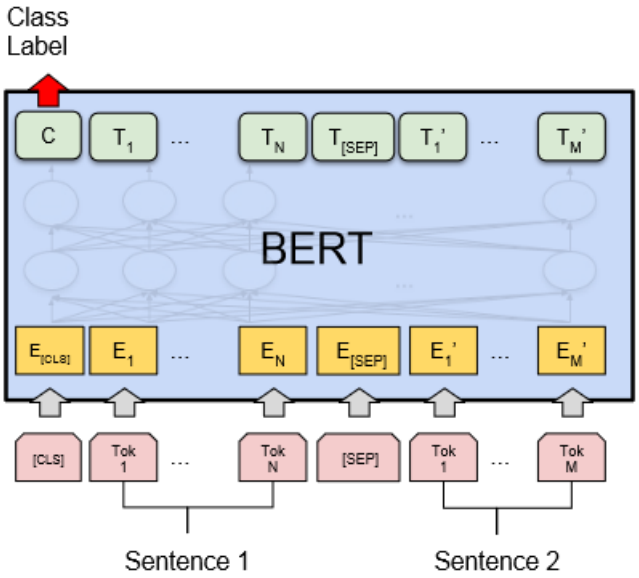


FIGURE 11: Architecture BERT (Devlin et al., 2018)

## 4.2 Augmentation de la quantité de données

À cause de la taille limitée de ce jeu de données et le déséquilibre des étiquettes, il est nécessaire de augmenter la quantité des données pour améliorer l'entraînement du modèle. Il existe plusieurs façon pour augmenter la quantité des données (voir la figure 12). Il existe plusieurs façons courantes comme l'extraction, la traduction inverse, l'insertion et la substitution.

- L'extraction : Extraire des exemples positifs ou négatifs en utilisant le modèle BM25 ou la pondération TF-IDF. Dans notre travail, les exemples positifs sont des phrases qui peuvent appuyer ou contredire une affirmation, tandis que les exemples négatifs sont des phrases sans rapport avec l'affirmation.

- La traduction inverse : par exemple, une phrase en français est d'abord traduite en allemand, puis retraduite en français.

- L'insertion : sélectionner un mot au hasard, choisir un de ses synonymes et l'insérer dans une position aléatoire dans la phrase

- La substitution : remplacer un mot par un synonyme (ou un antonyme) généré par WordNet ou un modèle BERT

Dans notre travail, nous exploitons l'extraction, la traduction inverse et la substitution pour augmenter la quantité des données. Puisque nous utilisons des données biomédicales, l'insertion d'un mot aléatoire pourrait changer le sens de la phrase, nous ne choisissons donc pas l'insertion.

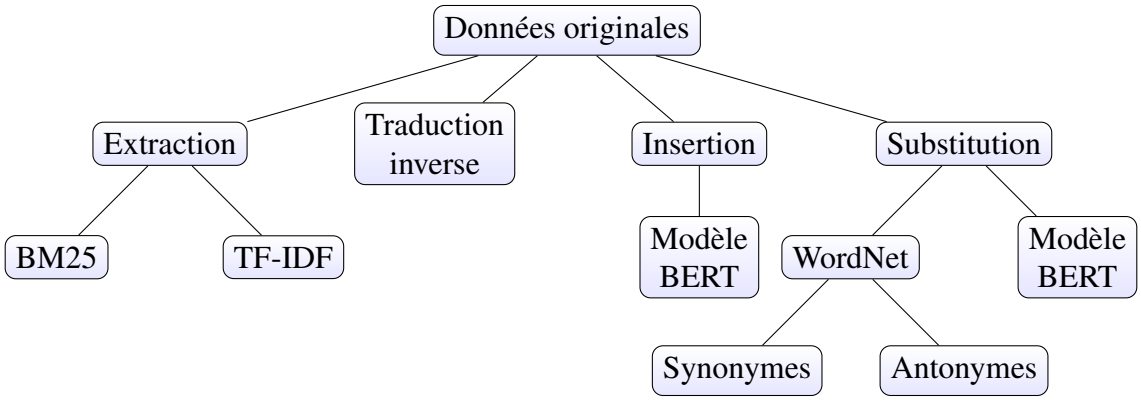


FIGURE 12: Augmentation des données

## 4.3 Modèles de langage

Concernant le choix de modèle, nous utilisons les modèles généraux comme RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2020) et MiniLM (Wang et al., 2020). Puisque notre recherche repose sur les données scientifiques dont la plupart sont liées au domaine biomédical, nous utilisons également les modèles entraînés sur des données scientifiques ou biomédicales tels que BioBERT (Lee et al., 2020), SciBERT (Beltagy et al., 2019) et Specter (Cohan et al., 2020).

### 4.3.1 Modèles pour le domaine général

#### RoBERTa

Au lieu d'utiliser le modèle BERT, nous choissison d'utiliser le modèle RoBERTa (Robustly Optimized BERT Approach), un variant de BERT proposé par Liu et al. (2019). Par rapport au BERT, RoBERTa est optimisé en termes de taille de modèle, de puissance de computation et de données d'entraînement.

#### DeBERTa

Créé par He et al. (2020), DeBERTa (Decoding-enhanced BERT with disentangled attention) utilise un nouveau mécanisme d'attention (disentangled attention mechanism) et renforce le décodeur, qui est plus puissant et aussi plus large que RoBERTa.

#### MiniLM

Les deux modèles précédemment mentionnés sont plus larges que les autres. Quand nous devons entraîner un tel modèle sur un jeu de données à grande échelle, nous avons besoin de plus de temps et des processeurs plus puissants. À cause de la limitation de la configuration, nous choisissons d'utiliser un modèle de petite taille. MiniLM est un modèle de langage proposé par Wang et al. (2020). Il est basé sur l'approche de distillation d'auto-attention profonde (Deep Self-Attention Distillation), permettant de transformer un grand modèle à base de Transformer en un modèle plus petit. Selon Wang et al. (2020), MiniLM est plus performant en comparaison avec les autres modèles plus petits que BERT, tels que DistillBERT et TinyBERT.

### 4.3.2 Modèles pour le domaine spéficique

#### BioBERT

BioBERT, proposé par Lee et al. (2020), est le premier modèle BERT entraîné sur des données biomédicales. Il est aussi l'un des modèles les plus utilisés pour le traitement des données biomédicales.

#### SciBERT

SciBERT est entraîné sur plus d'un million d'articles scientifiques, dont 82% sont issus du domaine biomédical et 18% du domaine informatique.

#### Specter

Specter est aussi un modèle de langage entraîné sur des articles scientifiques. Il est destiné à la génération de plongements de documents.

Dans ce chapitre, nous avons présenté les architectures et les modèles utilisées dans ce travail. Nous montrons en ci-dessous (voir figure 13) un schéma global des choix possibles pour les approches et les modèles qui peuvent nous aider à répondre à notre problématique et à créer notre système.

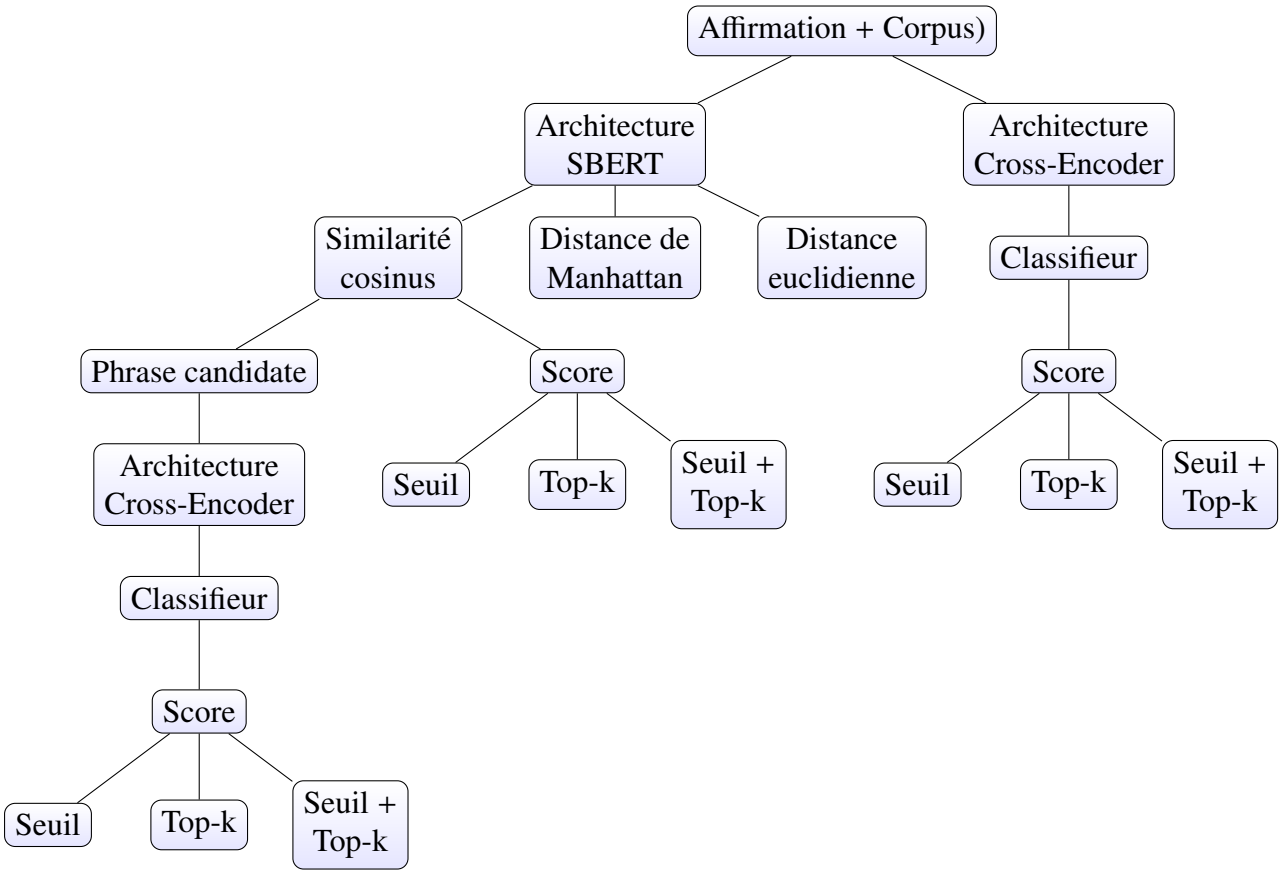


FIGURE 13: Combinaisons possibles des approches et des modèles pour résoudre notre problématique

# Chapitre 5. Expérimentations et résultats

## 5.1 Sélection de résumés

La sélection de résumés consiste à trouver des documents justificatifs à partir d'un corpus de résumés provenant d'articles scientifiques. Pour ce faire, nous exploitions tout d'abord l'architecture SBERT pour sélectionner les résumés candidats. Pour faciliter la génération de plongements de phrases, les phrases dans chaque résumé sont concaténées en une phrase. Selon la figure 14, nous utilisons deux modèles de langage pour générer respectivement la représentation vectorielle de l'affirmation et du résumé. Ces deux modèles peuvent être considérés comme identiques parce qu'ils partagent les mêmes configurations. Ensuite, nous appliquons l'opération de pooling pour obtenir une représentation de taille fixe qui permet de faciliter le traitement des représentations dans l'étape suivante. Enfin, nous calculons le cosinus pour comparer la similarité entre l'affirmation et le résumé, et choisissons les 30 résumés les plus pertinents en fonction du score de similarité.

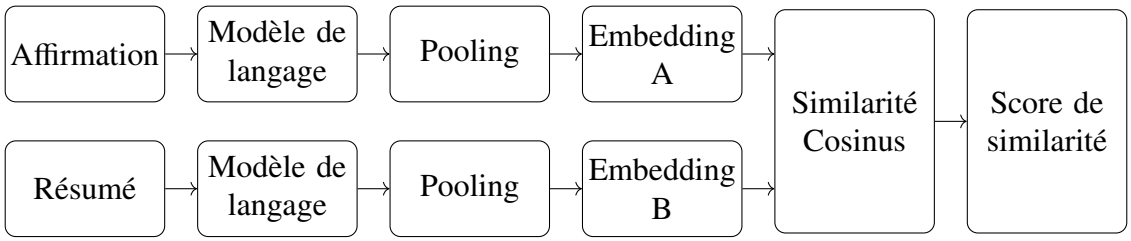


FIGURE 14: Architecture inspirée par (Devlin et al., 2018) pour la sélection de résumés

Pour améliorer la performance du système, nous recourons à l'architecture cross-encodeur proposé par Reimers and Gurevych (2019) pour évaluer les résumés sélectionnés à l'étape précédente. Comme montré dans la figure 15, le modèle prend l'affirmation et un des résumés sélectionnés en entrée et génère leur représentation vectorielle. Ensuite, un classifieur avec la fonction Sigmoid et BCE Loss (Binary Cross Entropy Loss) analyse la représentation générée et retourne une valeur entre 0 et 1 indiquant la similarité entre l'affirmation et le résumé.

Afin d'évaluer la performance de différents modèles, nous choisissons d'utiliser des modèles entraînés sur des données biomédicales ainsi que des modèle généraux. Pour le modèle entraîné sur un corpus spécialisé, nous utilisons BioBERT (Lee et al., 2020), SciBERT (Beltagy et al., 2019) et Specter (Cohan et al., 2020). Pour le modèle général, nous utilisons RoBERTa (Liu et al., 2019).

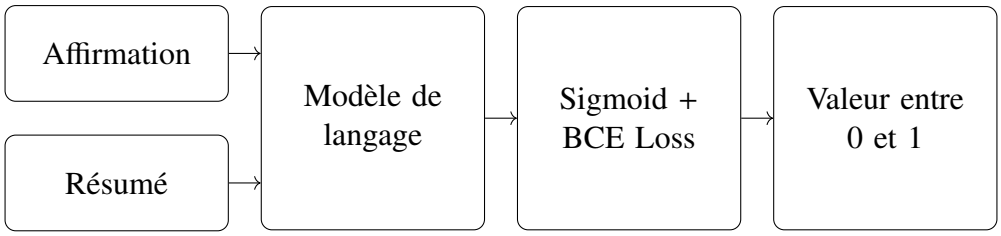


FIGURE 15: Architecture cross-encodeur inspirée par (Devlin et al., 2018) pour la sélection de résumés

En ce qui concerne l'entraînement du modèle, nous testons deux façons. Pour la première fois, nous utilisons les résumés pertinents comme exemples positifs et les autres résumés dans la liste « cited_doc_ids » comme exemples négatifs. Pour la deuxième fois, les exemples positifs restent les mêmes. Inspiré par le travail de (Pradeep et al., 2020), nous utilisons le modèle BM25 pour extraire des exemples négatifs. Nous utilisons Gensim. C'est une librairie Python qui propose l'implémentation du modèle BM25 et qui permet de mesurer la similarité sémantique entre phrases et de choisir les phrases candidates.

En général, le modèle BERT ne traite que les séquences d'une taille maximale de 512 tokens. Pour vérifier si cette restriction a une influence sur notre résultat, nous utilisons le modèle BERT pour générer le plongement de chaque phrase et calculons la moyenne des plongements de phrases. Le résultat montre que les scores sont plus faibles que précédemment. Cette situation peut s'expliquer par la figure 7, qui démontre que les phrases justificatives se trouvent souvent au début et au milieu du résumé.

| K retrieval | | 5 | | | |
|---|---|---|---|---|---|
| SBERT | Cross-encoder | Hit all | P | R | F1 |
| Specter | MINILM | 77.6 | 42.0 | 67.0 | 51.6 |

Tableau 10: Performance de la sélection de résumé sur des données de développement. Nous utilisons ici la moyenne des plongements de phrases dans le but de tester si la limitation de taille de séquence d'entrée de BERT a un impact sur le résultat.

Nous comparons dans la figure 16 le nombre de résumés pertinents et non pertinents avec leur score de similarité. L'axe horizontal est le score de similarité tandis que l'axe vertical est le nombre de phrases. Le score de similarité est calculé par le cosinus et varié entre 0 et 1. La plupart des résumés justificatives se trouvent dans l'intervalle de score de 0,8 à 1. Cela nous aide à fixer un seuil de 0,8 pour mieux sélectionner les résumés candidats.

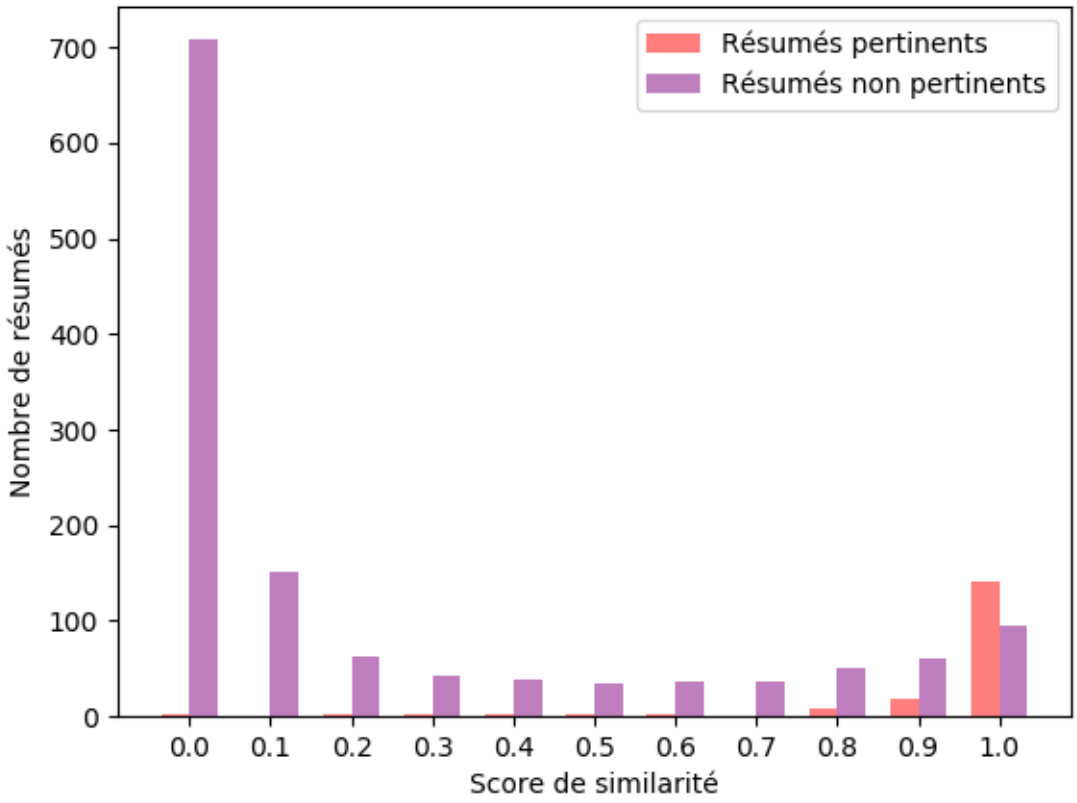


FIGURE 16: Comparaison du nombre de phrases justificatives et de phrases non justificatives

Les colonnes rouges signifient le nombre de résumés pertinents / justificatifs, et les colonnes violettes signifient le nombre de résumés non pertinents (qui ne peuvent pas soutenir ou contredire une affirmation). Ce graphique nous permet de fixer un seuil pour écarter les résumés inutiles dans la phase de sélection de résumés car il y a plus de résumés non pertinents que de résumés pertinents.

Dans le tableau 11, nous listons les modèles utilisés et les scores obtenus. La première colonne indique le modèle utilisé pour l'architecture SBERT. Nous mettrons ici le résultat de trois modèles : SciBERT, Specter et RoBERTa, et indiquons aussi le modèle utilisé pour augmenter la quantité des données (BM25 et TF-IDF). La deuxième colonne est le modèle utilisé pour l'architecture cross-encodeur. Les modèles SciBERT, Specter et RoBERTa sont tous entraînés sur le jeu de données SciFact, alors que le modèle MiniLM est entraîne sur MS MARCO, un jeu de données à grande échelle destiné au reclassement des textes. Pour mieux sélectionner les résumés justificatifs, nous exploitons la stratégie top-k. Pour l'évaluation, nous utilisons le score Hit all (le nombre de résumés correctement sélectionnés  le nombre attendu de résumés justificatifs), la précision, le rappel et le score F1 comme métrique. Nous pouvons constater que SciBERT et Specter sont plus performants que RoBERTa. Avec l'augmentation de la quantité des données, SciBERT confirme sa forte capacité de génération de plongements de phrases pour mesurer la similarité sémantique au niveau de document.

Nous comparons aussi dans le tableau 12 la stratégie top-k et la contrainte de seuil. Pour la sélection de résumés, le meilleur choix est de sélectionner les top-k documents.

| Top-k | | 3 | | | | 5 | | | |
|---|---|---|---|---|---|---|---|---|---|
| SBERT | Cross-encodeur | Hit all | P | R | F1 | Hit all | P | R | F1 |
| Système de référence (TF-IDF) | | 81.3 | 47.3 | 75.5 | 58.2 | 83.6 | 52.0 | 82.0 | 64.0 |
| SciBERT (BM25) | MiniLM | 90.9 | 57.0 | 90.9 | 70.0 | **93.0** | **57.3** | **91.4** | **70.4** |
| Specter (BM25) | MiniLM | 88.6 | 53.0 | 84.5 | 65.1 | 90.3 | 53.6 | 85.6 | 65.9 |
| Specter (TF-IDF) | MiniLM | 88.3 | 53.0 | 84.5 | 65.1 | 89.3 | 53.3 | 85.1 | 65.5 |
| Specter | MiniLM | 87.3 | 52.3 | 83.5 | 64.3 | 88.6 | 53.3 | 85.1 | 65.5 |
| SciBERT | MiniLM | 82.0 | 47.6 | 76.0 | 58.6 | 84.3 | 48.6 | 77.6 | 59.8 |
| | BM25 | 81.3 | 45.6 | 72.8 | 56.1 | 83.6 | 48.3 | 77.1 | 59.4 |
| RoBERTa Base (BM25) | MiniLM | 54.3 | 18.3 | 29.2 | 22.5 | 54.3 | 18.3 | 29.2 | 22.5 |

Tableau 11: Comparaison de différentes méthodes pour la sélection de résumés

Le système de référence est proposé par Wadden et al. (2020). Nous l'avons présenté dans la section 2.1.3

| SBERT | Cross-encoder | Top-k | Seuil | Hit all | P | R | F1 |
|---|---|---|---|---|---|---|---|
| Scibert (BM25) | MINILM | 5 | — | 93.0 | 57.3 | 91.4 | 70.4 |
| Scibert (BM25) | MINILM | — | 0.7 | 89.0 | 53.0 | 84.5 | 65.1 |
| Scibert (BM25) | MINILM | 5 | 0.7 | 88.3 | 52.6 | 84.0 | 64.7 |

Tableau 12: Comparaison de l'extraction des top-k résultats et la contrainte de seuil

## 5.2 Sélection de phrases

La deuxième étape est la sélection de phrases qui consiste à trouver des phrases justificatives à partir de résumés sélectionnés à l'étape précédente.

Dans cette phase, nous utilisons également le modèle de langage pour obtenir les présentations vectorielles de l'affirmation et de chaque phrase dans les résumés sélectionnés. Le système calcule la similarité entre eux et choisit les phrases les plus pertinentes en fonction du score de similarité retourné. Nous utilisons tout simplement l'architecture SBERT et les modèles SciBERT et Specter pour choisir les phrases justificatives. Nous construisons aussi un ensemble de données supplémentaires en utilisant la traduction inverse pour augmenter la quantité des données d'apprentissage. Selon la figure 13, contrairement à l'étape précédente, nous obtenons obtient un score F1 plus fort avec le modèle Specter. Malheureusement, notre système est moins performant que le système de référence, qui utilise le modèle RoBERTa-large entraîné sur deux jeux de données (SciFact et FEVER).

Nous listons aussi les scores obtenus à l'étape précédente. Quand nous utilisons le meilleur résultat obtenu à l'étape précédente comme entrée de la sélection de phrases (70,4), nous obtenons un score F1 plus faible que les autres (42,3). Cela nous fait comprendre que nous devons traiter ces deux étapes comme un ensemble et le système conçu doit être cohérent.

| Modèle | Sélection de résumés | | | Sélection de phrases | | |
|---|---|---|---|---|---|---|
| | | Hit all | F1 | P | R | F1 |
| Système de référence (RoBERTa Large) | Seuil = 0.7 | 89.0 | 65.1 | 50.1 | 59.2 | **54.3** |
| Système de référence (RoBERTa Large) | Top k = 5 | 93.0 | 70.4 | 35.9 | **61.7** | 45.4 |
| Specter | Seuil = 0.7 | 89.0 | 65.1 | **67.8** | 40.9 | 51.1 |
| Specter | Top k = 5, Seuil = 0.7 | 88.3 | 64.7 | 67.8 | 40.4 | 50.6 |
| Specter (Traduction inverse) | Seuil = 0.7 | 89.0 | 65.1 | 67.0 | 36.0 | 46.8 |
| Specter (Traduction inverse) | Top k = 5 | 93.0 | **70.4** | 49.0 | 37.1 | 42.3 |

Tableau 13: Résultat de la sélection des phrases sur l'ensemble de développement.

**Modèle conjoint**

Nous essayons aussi de combiner les deux premières étapes pour les traiter en même temps. Nous créons un système qui permet d'abord d'extraire les phrases les plus pertinentes puis de trouver à quels résumés elles appartiennent.

Pour entraîner le modèle, nous utilisons le modèle BM25 et la librairie Python NLPAug (Ma, 2019) qui permettent d'augmenter la quantité des données servant à sélectionner directement les phrases pertinentes à partir d'un ensemble de résumés. NLPAug nous propose plusieurs méthodes d'augmentation des données dont nous choisissons la substitution. Il génère des phrases synonymes en utilisant la base de données lexicale WordNet (Miller, 1995). Nous menons des expériences avec le modèle SciBERT sur les données d'apprentissage de SciFact. Selon le tableau 14, le modèle SciBERT est entraîné sur des données augmentées seulement avec BM25 et celles avec BM25 et WordNet. Le résultat est présenté dans le tableau 14. Les scores F1 ici sont plus faibles que ceux dans le tableau 13, donc nous n'avons pas adopté cette approche.

| Modèle | Augmentation des données | P | R | F1 |
|---|---|---|---|---|
| Scibert | BM25 | 32.7 | 34.9 | 33.8 |
| Scibert | BM25 + WordNet | 42.6 | 22.1 | 29.1 |

Tableau 14: Comparaison des scores des différentes méthodes d'augmentation de la quantité des données

## 5.3 Prédiction d'étiquettes

La prédiction d'étiquettes vise à reconnaître la relation d'implication entre l'affirmation et la justification, et attribuer une étiquette indiquant s'il y a une contradiction ou un accord entre eux.

La relation d'implication est divisée en trois classes : « SUPPORT », « CONTRADICT » et « NO_INFO ». En pratique, nous utilisons seulement les deux premières classes parce que les paires de phrases qui appartiennent à la dernière classe seront exclues à l'étape précédente. Nous construisons un système simple qui sert à choisir la classe la plus probable.

Le tableau 15 présente la comparaison des scores. Les métriques d'évaluation sont expliquées dans la section 3.3. Nous observons que les scores F1 obtenus avec notre système sont supérieurs à ceux obtenus avec le système de référence, sauf le score de la dernière métrique (Sentence+Label).

| | Abstract-level | | | | | | Sentence-level | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Label Only | | | Abstract + Rationale | | | Sentence Only | | | Sentence + Label | | |
| Model | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| Système de référence (RoBERTa Large) | 47.5 | 47.3 | 47.4 | 44.7 | 44.4 | 44.6 | 48.0 | 43.4 | 45.6 | 44.4 | 40.1 | **42.1** |
| DeBERTa XLarge | 59.3 | 44.0 | **50.5** | 52.9 | 39.2 | **45.0** | 50.4 | 42.0 | 45.9 | 43.6 | 36.3 | 39.6 |
| DeBERTa XXLarge | 57.0 | 42.5 | 48.7 | 50.6 | 37.7 | 43.2 | 51.6 | 42.3 | **46.5** | 43.0 | 35.2 | 38.7 |
| RoBERTa Large | 57.0 | 40.6 | 47.4 | 50.3 | 35.8 | 41.8 | 52.3 | 41.8 | **46.5** | 41.7 | 33.3 | 37.0 |
| BioBERT Large | 47.2 | 33.0 | 38.8 | 39.7 | 27.7 | 32.6 | 49.4 | 39.8 | 44.1 | 30.1 | 24.3 | 26.9 |

Tableau 15: Résultat de la prédiction d'étiquettes

Dans ce chapitre, nous avons mené des expériences sur le jeu de données SciFact et spécifié notre travail en trois étapes. Les résultats obtenus sont améliorables à cause de choix du modèle et de choix de l'approche. Nous allons discuter dans le chapitre suivant les limitations et les améliorations possibles.

# Chapitre 6. Discussion

Nous avons présenté dans le chapitre précédent comment vérifier une affirmation scientifique à l'aide de corpus et décrit un processus en trois étapes : sélection de résumés, sélection de phrases et prédiction d'étiquette. Nous avons évalué le module à chaque étape et obtenu un résultat final.

Avant d'analyser les problèmes et les limitations que nous avons rencontrés pendant le stage, nous voulons discuter le sens de justifier une affirmation scientifique. De nos jours, nous avons accès à de nombreuses informations grâce à Internet, notamment aux réseaux sociaux. Avec l'explosion de l'information, il est de plus en plus difficile de savoir ce qui est vrai ou faux. Dans le domaine de la recherche, en raison d'une grande quantité d'articles scientifiques, il est aussi difficile pour les chercheurs de se tenir au courant des nouvelles découvertes. Il est donc nécessaire de créer un outil informatique permettant de confirmer immédiatement la véracité des informations.

Pendant le stage, nous nous sommes rendu compte qu'il n'est pas si facile de créer un système pour justifier une affirmation scientifique. Tout d'abord, le corpus que nous avons utilisé est manuellement annoté. Beaucoup de facteurs peuvent affecter la qualité de l'annotation lors du travail d'annotation humaine, tel que la compréhension de l'article et de l'affirmation. Il est possible que l'annotateur donne des mauvaise étiquettes.

Ensuite, nous avons utilisé plusieurs modèles basés sur BERT en phase d'encodage de phrases. En général, le modèle BERT ne traite que les séquences de 512 tokens au maximum. Pour une séquence de plus de 512 tokens, il est obligatoire de la tronquer à 510 tokens (2 tokens réservés pour [CLS] et [SEP]). En phase de sélection de résumés, nous avons concaténé les phrases du résumé. La plupart des séquence d'entrée dépassent 512 tokens. Dans ce cas, le modèle ne traite que les 512 premiers tokens et néglige les autres.

Pour savoir si la limitation de 512 tokens affecte le résultat, nous avons essayé de calculer la moyenne des plongements de phrases. Nous avons également analysé la distribution des indices de phrase justificative afin de marquer les phrases les plus importantes du résumé pour cette tâche. Selon le tableau 11 présenté dans la section 5.1, cette limitation ne change pas le résultat puisque les phrases justificatives se trouvent souvent au début de résumé.

Il y a plusieurs façons pour traiter un texte de plus de 512 tokens. Par exemple, Sun et al. (2019) ont proposé trois méthodes pour tronquer un texte : (1) garder les 510 premiers tokens,

(2) garder les 510 derniers tokens ou (3) garder les 128 premiers et les 382 derniers. Ces trois méthodes sont évaluées sur les jeux de données IMDB et SOGOU. Selon les résultats, la troisième méthode est meilleure que les autres. Dans le futur, nous pourrons tronquer les longues séquences en gardant les 128 premiers et les 382 derniers. Nous prenons comme exemple le corpus que nous avons utilisé dans ce travail, où la plupart des résumes contiennent plus de 512 tokens. Pour améliorer la performance, nous pouvons tronquer chaque résumé en appliquant cette méthode.

De plus, nous pouvons utiliser le modèle de langage qui peut traiter un texte long comme Longformer. Longformer est proposé par Beltagy et al. (2020). C'est un variant de Transformer destiné au traitement de texte long (plus de 16000 tokens). En effet, selon les recherches précédentes (Pradeep et al. (2020), Li et al. (2020)), l'utilisation d'un modèle plus large sert à obtenir un score plus élevé dans l'évaluation, par exemple Pradeep et al. (2020) ont utilisé le modèle T5 et Li et al. (2020) ont utilisé RoBERTa-large. Dans notre étude, nous avons utilisé le modèle DeBERTa en phase de prédiction d'étiquettes et le résultat final nous montre l'efficacité de l'utilisation d'un modèle large.

Pour l'amélioration du système, nous pouvons exploiter d'autres approches. Par exemple, pour la sélection de résumés et la sélection de phrases, nous pouvons utiliser la distance de Manhattan ou la distance euclidienne pour calculer la similarité sémantique. Pour la prédiction d'étiquettes, nous pouvons utiliser le réseau LSTM et son variant comme ESIM, un LSTM renforcé pour reconnaître la relation d'implication.

# Chapitre 7. Conclusion et perspectives

Ce travail est consacré à la tâche de vérification d'affirmations dans la littérature scientifique. Dans ce mémoire, nous avons d'abord introduit dans le chapitre 2 un état de l'art des méthodes et des jeux de données les plus couramment utilisés pour trois tâches en rapport avec notre sujet : la vérification des faits, le calcul de similarité sémantique et l'inférence en langage naturel. Ensuite, nous avons décrit notre problématique et analysé le jeu de données utilisé dans ce travail dans le chapitre 3. Nous avons aussi présenté les métriques d'évaluation dans le même chapitre. Dans le chapitre 4, nous avons présenté deux architectures que nous avons utilisé pour construire notre système : l'architecture SBERT et l'architecture Cross-encodeur. Nous avons aussi présenté les modèles de langage utilisés.

Dans le chapitre 5, nous avons spécifié notre travail qui se décompose en trois étapes : la sélection de résumés, la sélection de phrases et la prédiction d'étiquettes. Pour la sélection de résumés, nous avons combiné les deux architectures SBERT et Cross-encodeur pour d'abord choisir les résumés candidats et puis les documents justificatives. Nous avons aussi utilisé la combinaison de deux architectures pour sélectionner les phrases justificatives, mais le résultat montre que l'utilisation d'une seule architecture SBERT est plus performante que la combinaison de deux architectures. Enfin, nous avons utilisé l'architecture Cross-encodeur pour reconnaître la relation d'implication entre l'affirmation et la phrase justificative et donner une étiquette.

Concernant le choix de modèle, nous avons utilisé les modèles généraux, tels que RoBERTa, DeBERTa, MiniLM, ainsi que les modèles entraînés sur des données scientifiques comme BioBERT, SciBERT et Specter. Puisque la taille des données est petite, nous avons appliqué les méthodes d'augmentation de la quantité des données. Par exemple, nous avons utilisé le modèle BM25 et la librairie NLPAug pour construire un jeu de données supplémentaire dont les exemples positifs sont des phrases justificatives et les exemples négatifs sont des phrases non pertinentes mais extraites du même résumés que les phrases justificatives.

En conclusion, les résultats ne sont pas très satisfaisants comme prévu à cause de la réalisation du système. Il nous reste encore beaucoup de choses à explorer et à améliorer. Par exemple, nous pouvons utiliser les approches d'apprentissage machine et les modèles de langage plus larges.

# Bibliographie


Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A. and Guo, W. (2013), * sem 2013 shared task : Semantic textual similarity, *in* 'Second joint conference on lexical and computational semantics (* SEM), volume 1 : proceedings of the Main conference and the shared task : semantic textual similarity', pp. 32–43.

Androutsopoulos, I. and Malakasiotis, P. (2010), 'A survey of paraphrasing and textual entailment methods', *Journal of Artificial Intelligence Research* **38**, 135–187.

Bahdanau, D., Cho, K. and Bengio, Y. (2014), 'Neural machine translation by jointly learning to align and translate', *arXiv preprint arXiv :1409.0473* .

Baroni, M. and Lenci, A. (2011), How we blessed distributional semantic evaluation, *in* 'Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics', pp. 1–10.

Beltagy, I., Lo, K. and Cohan, A. (2019), 'Scibert : A pretrained language model for scientific text', *arXiv preprint arXiv :1903.10676* .

Beltagy, I., Peters, M. E. and Cohan, A. (2020), 'Longformer : The long-document transformer', *arXiv :2004.05150* .

Bowman, S. R., Angeli, G., Potts, C. and Manning, C. D. (2015), 'A large annotated corpus for learning natural language inference', *arXiv preprint arXiv :1508.05326* .

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I. and Specia, L. (2017), 'Semeval-2017 task 1 : Semantic textual similarity-multilingual and cross-lingual focused evaluation', *arXiv preprint arXiv :1708.00055* .

Chandrasekaran, D. and Mago, V. (2021), 'Evolution of semantic similarity—a survey', *ACM Computing Surveys (CSUR)* **54**(2), 1–37.

Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H. and Inkpen, D. (2016), 'Enhanced lstm for natural language inference', *arXiv preprint arXiv :1609.06038* .

Cohan, A., Feldman, S., Beltagy, I., Downey, D. and Weld, D. S. (2020), SPECTER : Document-level Representation Learning using Citation-informed Transformers, *in* 'ACL'.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L. and Bordes, A. (2017), 'Supervised learning of universal sentence representations from natural language inference data', *arXiv preprint arXiv :1705.02364* .

Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S. R., Schwenk, H. and Stoyanov, V. (2018), 'Xnli : Evaluating cross-lingual sentence representations', *arXiv preprint arXiv :1809.05053* .

Derczynski, L., Bontcheva, K., Liakata, M., Procter, R., Hoi, G. W. S. and Zubiaga, A. (2017), 'Semeval-2017 task 8 : Rumoureval : Determining rumour veracity and support for rumours', *arXiv preprint arXiv :1704.05972* .

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018), 'Bert : Pre-training of deep bidirectional transformers for language understanding', *arXiv preprint arXiv :1810.04805* .

Disdier, C., Chalansonnet, M., Gagnaire, F., Gaté, L., Cosnier, F., Devoy, J., Saba, W., Lund, A. K., Brun, E. and Mabondzo, A. (2017), 'Brain inflammation, blood brain barrier dysfunction and neuronal synaptophysin decrease after inhalation exposure to titanium dioxide nano-aerosol in aging rats', *Scientific reports* **7**(1), 1–13.

Ferreira, W. and Vlachos, A. (2016), Emergent : a novel data-set for stance classification, *in* 'Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics : Human language technologies', pp. 1163–1168.

He, P., Liu, X., Gao, J. and Chen, W. (2020), 'Deberta : Decoding-enhanced bert with disentangled attention', *arXiv preprint arXiv :2006.03654* .

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M. and Blunsom, P. (2015), 'Teaching machines to read and comprehend', *Advances in neural information processing systems* **28**, 1693–1701.

Jones, K. S. (1972), 'A statistical interpretation of term specificity and its application in retrieval', *Journal of documentation* .

Kotlerman, L., Dagan, I., Szpektor, I. and Zhitomirsky-Geffet, M. (2010), 'Directional distributional similarity for lexical inference', *Natural Language Engineering* **16**(4), 359–389.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H. and Kang, J. (2020), 'Biobert : a pre-trained biomedical language representation model for biomedical text mining', *Bioinformatics* **36**(4), 1234–1240.

Levy, O., Remus, S., Biemann, C. and Dagan, I. (2015), Do supervised distributional methods really learn lexical inference relations ?, *in* 'Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies', pp. 970–976.

Li, X., Burns, G. and Peng, N. (2020), 'A paragraph-level multi-task learning model for scientific fact-verification', *arXiv preprint arXiv :2012.14500* .

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019), 'Roberta : A robustly optimized bert pretraining approach', *arXiv preprint arXiv :1907.11692* .

Liu, Y., Sun, C., Lin, L. and Wang, X. (2016), 'Learning natural language inference using bidirectional lstm model and inner-attention', *arXiv preprint arXiv :1605.09090* .

Ma, E. (2019), 'Nlp augmentation', https ://github.com/makcedward/nlpaug.

Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., Zamparelli, R. et al. (2014), A sick cure for the evaluation of compositional distributional semantic models., *in* 'Lrec', Reykjavik, pp. 216–223.

Miller, G. A. (1995), 'Wordnet : a lexical database for english', *Communications of the ACM* **38**(11), 39–41.

Nakashole, N. and Mitchell, T. (2014), Language-aware truth assessment of fact candidates, *in* 'Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)', pp. 1009–1019.

Nangia, N., Williams, A., Lazaridou, A. and Bowman, S. R. (2017), 'The repeval 2017 shared task : Multi-genre natural language inference with sentence representations', *arXiv preprint arXiv :1707.08172* .

Negre, E. (2013), 'Comparaison de textes : quelques approches...'.

Popat, K., Mukherjee, S., Strötgen, J. and Weikum, G. (2017), Where the truth lies : Explaining the credibility of emerging claims on the web and social media, *in* 'Proceedings of the 26th International Conference on World Wide Web Companion', pp. 1003–1012.

Popat, K., Mukherjee, S., Yates, A. and Weikum, G. (2018), 'Declare : Debunking fake news and false claims using evidence-aware deep learning', *arXiv preprint arXiv :1809.06416* .

Pradeep, R., Ma, X., Nogueira, R. and Lin, J. (2020), 'Scientific claim verification with vert5erini', *arXiv preprint arXiv :2010.11930* .

Rashkin, H., Choi, E., Jang, J. Y., Volkova, S. and Choi, Y. (2017), Truth of varying shades : Analyzing language in fake news and political fact-checking, *in* 'Proceedings of the 2017 conference on empirical methods in natural language processing', pp. 2931–2937.

Reimers, N. and Gurevych, I. (2019), Sentence-bert : Sentence embeddings using siamese bert-networks, *in* 'Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics.
**URL:** *https ://arxiv.org/abs/1908.10084*

Robertson, S., Zaragoza, H. and Taylor, M. (2004), Simple bm25 extension to multiple weighted fields, *in* 'Proceedings of the thirteenth ACM international conference on Information and knowledge management', pp. 42–49.

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiskỳ, T. and Blunsom, P. (2015), 'Reasoning about entailment with neural attention', *arXiv preprint arXiv :1509.06664* .

Rush, A. M., Chopra, S. and Weston, J. (2015), 'A neural attention model for abstractive sentence summarization', *arXiv preprint arXiv :1509.00685* .

Song, B., Liu, J., Feng, X., Wei, L. and Shao, L. (2015), 'A review on potential neurotoxicity of titanium dioxide nanoparticles', *Nanoscale research letters* **10**(1), 1–17.

Sun, C., Qiu, X., Xu, Y. and Huang, X. (2019), How to fine-tune bert for text classification ?, *in* 'China National Conference on Chinese Computational Linguistics', Springer, pp. 194–206.

Syed, Z. H., Röder, M. and Ngonga Ngomo, A.-C. (2018), Factcheck : Validating rdf triples using textual evidence, *in* 'Proceedings of the 27th ACM International Conference on Information and Knowledge Management', pp. 1599–1602.

Thorne, J. and Vlachos, A. (2018), 'Automated fact checking : Task formulations, methods and future directions', *arXiv preprint arXiv :1806.07687* .

Thorne, J., Vlachos, A., Christodoulopoulos, C. and Mittal, A. (2018), 'Fever : a large-scale dataset for fact extraction and verification', *arXiv preprint arXiv :1803.05355* .

Turney, P. D. and Mohammad, S. M. (2015), 'Experiments with three approaches to recognizing lexical entailment', *Natural Language Engineering* **21**(3), 437–476.

Vlachos, A. and Riedel, S. (2014*a*), Fact checking : Task definition and dataset construction, *in* 'Proceedings of the ACL 2014 workshop on language technologies and computational social science', pp. 18–22.

Vlachos, A. and Riedel, S. (2014*b*), Fact checking : Task definition and dataset construction, *in* 'LTCSS@ACL'.

Vlachos, A. and Riedel, S. (2015), Identification and verification of simple claims about statistical properties, *in* 'Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, pp. 2596–2601.

Vlachos, M. (2010), *Similarity Measures*, Springer US, Boston, MA, pp. 903–906.
**URL:** *https ://doi.org/10.1007/978-0-387-30164-8*$_7$60

Vo, N. and Lee, K. (2020), 'Where are the facts ? searching for fact-checked information to alleviate the spread of fake news', *arXiv preprint arXiv :2010.03159* .

Wadden, D., Lo, K., Wang, L. L., Lin, S., van Zuylen, M., Cohan, A. and Hajishirzi, H. (2020), 'Fact or fiction : Verifying scientific claims', *arXiv preprint arXiv :2004.14974* .

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N. and Zhou, M. (2020), 'Minilm : Deep self-attention distillation for task-agnostic compression of pre-trained transformers', *arXiv preprint arXiv :2002.10957* .

Wang, W. Y. (2017), '" liar, liar pants on fire" : A new benchmark dataset for fake news detection', *arXiv preprint arXiv :1705.00648* .

Zhang, K., Chen, E., Liu, Q., Liu, C. and Lv, G. (2017), A context-enriched neural network method for recognizing lexical entailment, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 31.

# Table des figures

# Liste des tableaux

**MOTS-CLÉS** : TAL, vérification d'affirmations scientifiques, inférence en langage naturel

## RÉSUMÉ

La vérification d'affirmations scientifiques est une nouvelle tâche qui est apparue récemment dans le domaine du TAL. Elle consiste à confirmer la véracité d'une affirmation et à trouver des justifications à partir d'un corpus d'articles scientifiques. La justification peut être un texte ou une phrase extraite de corpus, qui appuie ou contredit l'affirmation. Dans ce mémoire, nous avons exploité deux architectures pour concevoir un système destiné à la vérification d'affirmations et à la prédiction d'étiquettes. Nous avons construit un système qui traite cette tâche en trois étapes : d'abord nous sélectionnons les documents justificatifs à partir d'un corpus d'articles scientifiques, ensuite nous choisissons les phrases justificatives et enfin nous donnons une étiquette indiquant les phrases justificatives qui appuient ou contredisent l'affirmation.

**KEYWORDS** : NLP, scientific claim verification, natural language inference

## ABSTRACT

Scientific claim verification is a new task that has recently emerged in NLP. It aims to verify the truthfulness of a scientific claim and find evidence in a scientific corpus. The evidence could be a text, or a sentence extracted from the corpus, which supports or contradicts the claim. In this paper, we use two architectures to design our system for claim verification and label prediction. The task is divided into three steps: first we retrieve the evidence abstracts from the corpus, then we select the evidence sentences from the retrieved abstracts and finally we label the evidence sentences as supporting or contradicting the claim.