Interprétation de prédictions de modèles pour l'inférence textuelle en perturbant significativement les entrées

Marjorie Armando

Laboratoire d'Informatique et des Systèmes - LIS

Encadrant

Benoit Favre

Résumé

La reconnaissance de l'inférence textuelle (Recognizing Textual Entailment : RTE) est au coeur de tous les aspects de la compréhension de texte en traitement automatique du langage (TAL). Le but de la RTE est de déterminer automatiquement si une phrase, appelée l'hypothèse, peut être déduite d'une autre phrase, appelée la prémisse. En utilisant des réseaux de neurones, nous pouvons obtenir de bons taux de réussite pour la RTE. Cependant, ces réseaux ne sont pas interprétables, ainsi nous n'avons pas la possibilité de savoir si le modèle s'est basé sur de bonnes informations pour sa décision. Dans ce travail, nous proposons la méthode Best Adversarial eXample for Interpretability (BAXI), qui respecte les règles d'une "bonne" explication pour rendre un modèle interprétable dans le cadre de la RTE, avec l'utilisation du corpus SNLI. Nous allons la comparer à la méthode Local Interpretable Model-agnostic Explanations (LIME) qui permet d'expliquer les prédictions de n'importe quel classifieur en apprenant localement un modèle interprétable dans le voisinage de l'entrée. Pour cela, nous avons extrait un échantillon du corpus de test de SNLI, et demandé à six annotateurs de donner les mots expliquant l'étiquette associée à la paire prémisse/hypothèse. Grâce à ce nouveau corpus, nous mesurons le taux d'explications correctes fourni par la méthode BAXI et la méthode LIME, sur plusieurs systèmes que nous avons implémentés.

Mots-clés : inférence textuelle, apprentissage automatique,
traitement automatique des langues naturelles, interprétabilité, LSTM

1 Introduction

1.1 Contexte et motivations

Dans la pratique, il y a souvent un compromis entre le taux de réussite et l'interprétabilité d'un modèle. Certains modèles tels que les arbres de décisions ou encore les modèle linéaires sont facilement interprétables, et sont donc parfois utilisés à la place de modèles complexes tels que les réseaux de neurones profonds, même si ceux-ci peuvent donner de meilleurs résultats.
Pouvoir interpréter un modèle permettrait de le rendre utilisable dans des domaines où les décisions doivent être mûrement réfléchies, telle que la médecine. Si un modèle propose de donner un certain traitement à un patient, il faut que le modèle puisse donner de bonnes explications pour que le docteur l'approuve, car les conséquences pourraient être catastrophiques.

Dans le cadre de la reconnaissance de l'inférence textuelle (Recognizing Textual Entailment : RTE), les modèles obtenant les meilleurs taux de réussite à l'heure actuelle sont non interprétables. Le problème avec l'utilisation de modèles non interprétables est qu'ils fournissent uniquement les probabilités de chaque étiquette. Comment savoir, de manière humainement compréhensible, pourquoi le modèle a associé une étiquette particulière à une certaine entrée ?

Pourtant, avoir un modèle digne de confiance lui permettrait d'être d'avantage utilisé : en effet, il a été observé, par exemple, que le fait de fournir des explications augmente l'acceptation des recommandations de films [1]. De plus, cela permettrait également au développeur de choisir un modèle parmi ceux qu'il a implémenté, car le taux de réussite n'est pas le seul critère à prendre en compte : un modèle peut fournir l'étiquette attendue en se basant sur de mauvaises informations. Pour qu'un modèle soit digne de confiance, il faut qu'un grand nombre d'explications fournies soit jugé correct par les utilisateurs.

1.2 Tâches

Les concepts sémantiques d'inférence et de contradiction sont au coeur de tous les aspects de la compréhension de texte en TAL. Ainsi, la RTE est essentielle dans des tâches telles que la recherche d'information, le raisonnement de bon sens, ou encore les systèmes de question/réponse.
Plus spécifiquement, soit une paire de phrase prémisse/hypothèse, la RTE a comme objectif de détecter si la seconde phrase est en contradiction, peut être déduite ou bien est neutre par rapport à la première phrase. Il y a donc trois étiquettes permettant d'illustrer la relation entre la prémisse et l'hypothèse :

- Contradiction : l'hypothèse contredit la prémisse. Par exemple :
  Prémisse : "Le chat est entièrement blanc."
  Hypothèse : "Le chat est entièrement noir."

- Neutre : l'hypothèse est possible dans le contexte de la prémisse.
  Par exemple :
  Prémisse : "Le chat dort sur la banquette."
  Hypothèse : "Le chat aime le chocolat."

- Inférence : l'hypothèse est déduite de la prémisse. Par exemple :
  Prémisse : "Le chat aimerait manger la souris."
  Hypothèse : "Le chat a faim."

Dans ce travail, nous proposons d'interpréter les prédictions des modèles pour la RTE avec la méthode Best Adversarial eXample for Interpretability (BAXI). Nous allons comparer ses performances avec la méthode Local Interpretable Model-agnostic Explanations (LIME) [1], grâce à un corpus explicatif que nous avons créé à l'aide de six annotateurs.

Nous allons tout d'abord lister nos hypothèses scientifiques en section 2, puis énoncer l'état de l'art en section 3 pour la RTE ainsi que pour l'interprétabilité de prédictions de modèle en général. Nous verrons l'approche de LIME en section 3.3. Nous allons ensuite décrire ce qu'est une "bonne" explication pour savoir sur quoi faut-il se baser pour implémenter une méthode d'interprétabilité en section 4. Puis, nous décrirons les modèles que nous avons implémentés pour la RTE en section 5. Ces modèles sont les boîtes noires que nous tenterons d'interpréter. Enfin, nous allons décrire la méthode BAXI en section 6 puis le cadre expérimental en section 7 pour donner les différents résultats obtenus en section 8. Nous conclurons sur une discussion et sur les différentes perspectives en section 9 et 10.

2 Hypothèse scientifique

Notre principale hypothèse étudiée lors de ce travail est que BAXI repère de meilleurs mots explicatifs que LIME pour la tâche de la RTE : BAXI $\underset{RTE}{>}$ LIME.
Un mot explicatif est un mot ayant une grande importance pour l'étiquette considérée. La comparaison est effectuée à l'aide d'un corpus explicatif contenant les mots explicatifs dans la prémisse et dans l'hypothèse (voir section 7.2).

3 Etat de l'art

3.1 Dans la RTE

Les premiers travaux sur la RTE ont été formé sur de très petits ensembles de données avec des méthodes conventionnelles, telles que les méthodes peu profondes [2] ou encore les méthodes de logique naturelle [3].
Ces dernières années, la tâche de la RTE a connu une nette amélioration, en particulier grâce à la publication du corpus SNLI (the Stanford Natural Language Inference) décrit dans la section 7.1 [4]. Cela a permit d'entraîner des réseaux de neurones complexes qui nécessitent une quantité relativement importante de données. En effet, la recherche sur l'apprentissage automatique dans ce domaine été limitée par le manque de corpus assez grand.

/*DECRIRE UNE APPROCHE*/

3.2 Dans l'interprétabilité de prédictions de modèles

Plusieurs méthodes existent pour l'interprétation de prédictions de modèles, mais toutes ne s'appliquent pas au texte. La méthode de Fong et Vedaldi [5] en est un exemple : les auteurs perturbent une image en appliquant un masque par dessus pour flouter quelques parties, puis voient si la probabilité de l'étiquette initiale a baissé. Si c'est le cas, alors l'objet flouté dans l'image est un facteur explicatif à la décision du modèle.

D'autres méthodes d'interprétation existent, telle que la méthode Deep Learning Important FeaTures (DeepLIFT) pour l'apprentissage profond, qui décompose la prédiction d'un réseau de neurones sur une entrée spécifique en rétropropageant les contributions de tous les neurones du réseau à chaque *feature* de l'entrée. DeepLIFT compare l'activation de chaque neurone à son "activation de référence" et attribue des scores de contribution en fonction de la différence. L'activation de référence est choisie par l'utilisateur [6].
On peut également citer la méthode de propagation de pertinence par couche pour l'interprétation des réseaux de neurones profonds, une méthode équivalente à DeepLIFT avec l'activation de référence de tous les neurones fixée à 0 [7].

Une autre approche est la méthode SHapley Additive exPlanations (SHAP) [8] : elle explique la prédiction de n'importe quel modèle en utilisant les valeurs de Shapley, introduites dans la théorie des jeux coopératives en 1953 [9].
SHAP attribue des paiements aux joueurs en fonction de leur contribution au paiement total. Les joueurs coopèrent dans une coalition et obtiennent un certain gain de cette coopération.
Ici, le "jeu" est la tâche de prédiction pour une seule instance. Le "gain" est la prédiction réelle pour cette instance moins la prédiction moyenne de toutes les instances. Les "joueurs" sont les valeurs des *features* présentes dans l'instance (formant une coopération), qui collaborent pour recevoir le gain. La valeur de Shapley d'une valeur pour une *feature* est la contribution marginale moyenne de cette valeur de *feature* sur toutes les coopérations de *features* possibles.

Ces valeurs ont récemment été utilisées pour attribuer une mesure d'importance aux *features* [10]. SHAP appartient à la classe des "méthodes d'attribution de *features* additives" : elle attribue une valeur à chaque valeur de *features* pour chaque prédiction (c'est-à-dire une attribution de *feature*). Plus la valeur est haute, plus l'attribution de la *feature* à la prédiction spécifique est grande. Un bon exemple est décrit dans le livre *"Interpretable Machine Learning"* de Molnar dans le chapitre 5.7 [10].
SHAP est la seule méthode qui respecte les trois propriétés suivantes, ce qui en fait la seule méthode basée sur une théorie solide :

- Précision locale : les explications sont fidèles et véridiques au modèle.
- *Feature* manquante : les *features* retirées n'ont aucun impact attribué aux prédictions du modèle.
- Cohérence : les explications sont cohérentes avec l'intuition humaine.

Techniquement, la cohérence indique que si un modèle change de sorte que la contribution d'une entrée augmente ou reste la même indépendamment des autres entrées, l'attribution de cette entrée ne devrait pas diminuer.

Le calcul de ces valeurs est cependant très coûteux : il faut générer toutes les coopérations possibles pour une instance, ce qui en fait une complexité exponentielle et n'est donc pas utilisable dans la pratique. Un algorithme efficace a été mis au point par les auteurs de SHAP, cependant il fonctionne uniquement sur les modèles basés sur les arbres.

Concernant l'interprétabilité dans la tâche de la RTE, ce n'est que très récemment qu'une méthode a été proposée, mais uniquement pour expliquer l'inférence [11]. Cette méthode entraîne un classifieur pour qu'il puisse apprendre automatiquement la relation sémantique entre les mots de la prémisse et l'hypothèse grâce à WordNet. Pour interpréter la décision de l'inférence, il faut naviguer dans le graphe WordNetGraph, créé par les auteurs à partir des définitions de WordNet. Voici un exemple :

Prémisse : "Many cellphones have built-in digital cameras."
Hypothèse : "Many cellphones can take pictures."

Premièrement, la méthode recherche des paires de mots qui ont une forte relation sémantique et qui peuvent prouver que cette inférence est vraie, puis ces paires sont envoyées en entrée à l'algorithme de navigation graphique. Dans cet exemple, la meilleure paire est "*digital camera*" (la source) et "*pictures*" (la cible). En partant de la source, la méthode récupère tous les noeuds de WordNetGraph, et calcule la similarité sémantique entre chaque noeud et la cible et choisit le noeud qui a la valeur la plus élevée comme prochain noeud à visiter. Cela est fait récursivement jusqu'à atteindre la cible. Les segments suivants sont trouvés par l'algorithme de navigation :

<digital camera has_ supertype camera>
<camera has_ supertype equipment>
<equipment has_ diffqual for taking photographs>

où le premier segment signifie que "*digital camera*" appartient à la classe "*camera*", le deuxième segment signifie que "*camera*" appartient à la classe "*equipment*", et le dernier segment signifie que "*equipment*" a la qualité de pouvoir prendre des photographies ("*for taking photographs*").
"*photographs*" et "*pictures*" sont synonymes, donc la recherche dans le graphe s'arrête ici, et confirme bien l'inférence. Les explications suivantes, construites à partir des segments, sont alors données :

A digital camera is a kind of camera
A camera is an equipment for taking photographs
Photograph is synonym of picture

3.3 Description de LIME

Nous étudions dans ce travail la méthode LIME qui permet d'expliquer les prédictions de n'importe quel classifieur ou regresseur. L'objectif global de LIME

est d'identifier un modèle interprétable parmi le voisinage de l'entrée x.

Tout d'abord, les *features* utilisées et les représentations interprétables des *features* sont à distinguer. Par exemple, les *features* sont les *embeddings* des mots et la représentation interprétable de ces *features* est un vecteur binaire qui indique la présence ou l'absence des mots.

LIME définit une explication par un modèle $g \in G$, où G est la classe des modèles interprétables tels que les modèles linéaires ou les arbres de décisions. Vu que les modèles interprétables n'ont pas tous la même difficulté à être interprété, LIME définit $\Omega(g)$ qui est une mesure de la complexité d'interpréter g. En prenant l'example des arbres de décisions, $\Omega(g)$ est la profondeur.
On dénote par $f : \mathbb{R}^d \to \mathbb{R}$ le modèle utilisé comme une boîte noire. $f(x)$ est la probabilité que l'entrée x appartienne à une certaine étiquette.
LIME va alors se baser sur la représentation interprétable des données en retirant un ou plusieurs mots au hasard. Cette nouvelle entrée est notée z. LIME définit la localité de x avec $\pi_x(z)$ qui est une mesure de proximité entre z et x. C'est un noyau se basant sur la similarité cosinus.
Enfin, LIME définit $\mathcal{L}(f, g, \pi_x)$ qui est une mesure pour savoir à combien g est infidèle à f dans la localité défini par π_x. Pour préserver à la fois l'interprétabilité et la fidèlité locale, LIME minimise $\mathcal{L}(f, g, \pi_x)$ avec $\Omega(g)$ assez petit pour être interprétable par les humains. L'explication de LIME est donc la suivante :

$$\mathcal{E}(x) = \underset{g \in G}{argmin} \quad \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{1}$$

Cette formule peut être utilisée par différents modèles $g \in G$, fonctions de fidèlité $\mathcal{L}(f, g, \pi_x)$, et mesure de complexité $\Omega(g)$.

LIME peut alors donner les K mots les plus importants de l'entrée x pour toutes étiquettes. La figure 1 est un exemple illustant le principe de LIME. Concernant l'évaluation des explications de LIME pour savoir si la méthode fournit des explications correctes, les auteurs ont utilisé LIME /*A FINIR */.

4 Définition d'interprétabilité

Il n'y a malheureusement pas de consensus concernant la définition d' "interprétabilité". Miller définit cela comme étant le degré auquel un humain peut comprendre la cause d'une décision [12]. Un système a donc une meilleure interprétabilité qu'un autre si ses explications sont plus faciles à comprendre par un humain.

4.1 Qu'est-ce-qu'une explication ?

La définition donnée par Miller est assez simple : une explication est une réponse à une question commençant par "pourquoi". Une question commençant par "comment" peut être retournée en une question commençant par "pourquoi". Le terme "explication" désigne le processus social et cognitif d'expliquer, mais c'est également le produit de ces processus.

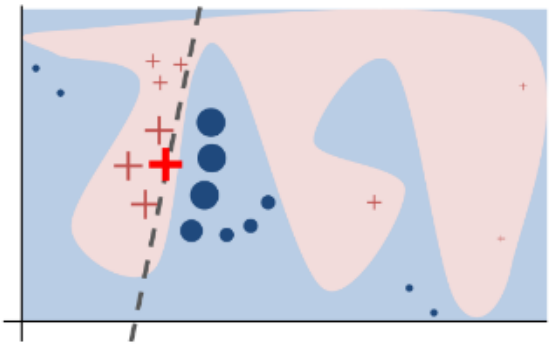

FIGURE 1 – **Exemple présentant l'intuition de LIME. La décision de la boîte noire f, inconnu par LIME, est représentée par le fond bleu et rose. La croix rouge en gras est l'entrée x que l'on veut expliquer. LIME crée des entrées modifiées, utilise f pour avoir la probabilité de ces entrées pour l'étiquette y, et les pondère par leur proximité par rapport à x (les poids sont représentés par la taille). La droite pointillée est l'explication apprise qui est localement fidèle. [1]**

4.2 Qu'est-ce-qu'une "bonne" explication ?

La définition d'une bonne explication ne doit pas se baser sur l'intuition de l'auteur, mais plutôt sur des faits. Miller résume ce qu'est une bonne explication, c'est-à-dire ce que les humains attendent d'une explication [10]. BAXI doit suivre les trois règles suivantes, caractérisant une bonne explication :

Explication contrastée C'est une explication qui doit être comparée. Les utilisateurs se demandent généralement pourquoi cette prédiction a été faite et pas une autre, via la question "quelle aurait été la prédiction si cette entrée avait été changé par une autre ?". Un docteur se demandant "pourquoi ce traitement ne marche pas sur ce patient ?" voudrait comparer les données de ce patient à un autre patient ayant des caractéristiques similaires mais pour qui le traitement marche. La meilleure explication pour ce type d'explication est celle qui met en évidence les différences entre l'entrée traitée et l'entrée de comparaison. L'entrée de comparaison peut être artificielle.

Explication sélective C'est une explication qui doit être courte. Généralement, un phénomène s'explique par plusieurs facteurs. Il faut en donner peu, à savoir deux ou trois raisons, même si les explications peuvent être plus complexes que cela.

Explication compréhensible Comme nous l'avons expliqué ci-dessus, une explication est un processus social, c'est-à-dire qu'il faut prendre en compte les connaissances de la personne à qui l'on veut donner une explication. Dans notre travail, nous partons du principe qu'une explication doit être comprise par tout le monde, que ce soit par un expert du domaine de l'apprentissage automatique ou bien par quelqu'un qui n'en a jamais entendu parler. Si l'explication est

complexe, on pourrait la structurer comme le fait la méthode de Vivian et al. décrite en section 3.2 [11].

5 Approches pour la RTE

Réseaux de neurones récurrent Nous avons implémenté trois systèmes différents avec la librairie DyNet [13]. Les entrées -les mots de la prémisse et de l'hypothèse- sont représentées par des *embeddings* et les sorties sont les probabilités de chaque étiquette de la RTE. Nous utilisons les Long Short-Term Memory (LSTMs) et les Bidirectionnal Long Short-Term Memory (BILSTMs). Ils permettent de relier des informations vues antérieurement à la tâche courante. On leur donne des informations à chaque instant t, créant ainsi une cellule.

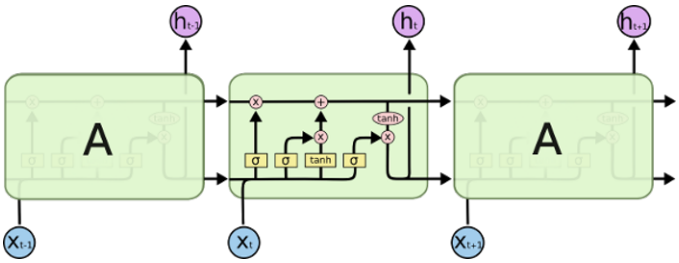

FIGURE 2 – **Schéma d'un LSTM contenant trois cellules. Les différentes portes sont représentées dans la cellule du milieu. Les entrées sont les mot X_{t-1}, X_t et X_{t+1}. Les états cachés calculés sont h_{t-1}, h_t et h_{t+1}.**

Ce sont des réseaux de neurones récurrent particuliers capable d'apprendre une dépendance très éloignée à notre tâche courante : ils ont la capacité de supprimer ou d'ajouter des informations à l'état de la cellule, régulé par des portes qui décident s'il faut laisser passer de l'information.
Ces réseaux de neurones sont utilisés ici en tant qu'encodeur : on fournit en entrée, à chaque instant t, l'embedding du t$^{\text{ième}}$ mot de la phrase traitée, à savoir la prémisse ou l'hypothèse. Le réseau calcule alors à chaque instant t un état caché h_t comme suit pour chaque mot :

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{3}$$
$$u_t = \tanh(W_u x_t + U_u h_{t-1} + b_u) \tag{4}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{5}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot u_t \tag{6}$$
$$h_t = o_t \odot \tanh(c_t) \tag{7}$$

où σ est la fonction sigmoid, $\odot$ est le produit de Hadamard entre deux vecteurs, W_i, W_f, W_u, $W_o \in \mathbb{R}^{D \times D_e}$, U_i, U_f, U_u, $U_o \in \mathbb{R}^{D \times D}$ et b_i, b_f, b_u, $b_o \in \mathbb{R}^D$ sont des paramètres mis-à-jour par le réseau. D est la dimension des états cachés et D_e est la dimension des *embeddings*. Un LSTM peut donc générer une représentation cachée d'un mot (en violet dans la figure 2), et une représentation

cachée d'une phrase qui est l'état caché de la dernière cellule. Un BILSTM est quant à lui un LSTM bidirectionnel qui exploite les mots dans l'ordre naturel et l'ordre inversé.

Premier système Le premier système passe la prémisse et l'hypothèse en entrée d'un LSTM pour avoir une représentation pour chacune de ces deux phrases. On les concatène pour les envoyer ensuite à une couche de décision :

$$y = softmax(W \times [LSTM(prémisse)\ ; LSTM(hypothèse)] + b) \tag{8}$$

où y est un vecteur contenant les probabilités de chaque étiquette, W est la matrice de poids, $LSTM(prémisse)$ et $LSTM(hypothèse)$ sont respectivement les représentations issues du LSTM entraînés avec le reste du modèle de la prémisse et de l'hypothèse, b est le biais, et $[;]$ dénote la concaténation.

Deuxième système Le deuxième système effectue le même mécanisme que le premier système pour avoir une représentation de la prémisse et de l'hypothèse. On compare les deux représentations pour envoyer la comparaison à une couche de décision :

$$y = softmax(W \times (LSTM(prémisse) \times LSTM(hypothèse)^T) + b) \tag{9}$$

où T dénote la transposée. Ce système permet de comparer les phrases, contrairement au système 1 qui les concatène.

Troisième système Le troisième système est inspiré de la méthode KIM [14]. On représente les mots de la prémisse et de l'hypothèse en les passant dans un BILSTM qui utilise un LSTM *forward* pour lire la phrase de gauche à droite, puis un LSTM *backward* pour lire la phrase dans l'autre sens. A chaque mot lu, un état caché est généré par les deux LSTMs. Ces deux états cachés sont alors concaténés pour obtenir une représentation du mot :
$h_t = [h_t^{\rightarrow}; h_t^{\leftarrow}]$, où $h_t^{\rightarrow}$ est l'état caché généré par le LSTM *forward* à l'instant t, $h_t^{\leftarrow}$ est celui généré par le LSTM *backward* à l'instant t, et h_t est la représentation du mot t.
On dénote par p^s (respectivement h^s) le vecteur de représentation des mots de la prémisse (respectivement de l'hypothèse). Nous construisons ensuite une matrice d'alignement comme suit :

$$e_{ij} = (p_i^s)^T h_j^s \tag{10}$$

où p_i^s est la représentation du i$^{\text{ème}}$ mot de la prémisse, h_j^s est celle du j$^{\text{ième}}$ mot de l'hypothèse. Avec cette matrice, nous pouvons alors construire les vecteurs de contexte p^c et h^c suivants pour la prémisse et l'hypothèse :

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{N} exp(e_{ik})}, p_i^c = \sum_{j=1}^{N} \alpha_{ij} h_j^s \tag{11}$$

$$\beta_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{M} exp(e_{kj})}, h_j^c = \sum_{i=1}^{M} \beta_{ij} p_i^s \tag{12}$$

Système	Inférence	Neutre	Contradiction	Dev
1	73,90%	68,38%	67,33%	69,89%
2	**87,98%**	**74,99%**	**78,55%**	**80,57%**
3	74,92%	69,55%	68,73%	71,09%

TABLE 1 – **Résultats des taux de réussite en pourcentage des étiquettes prédites pour le corpus de validation de SNLI, par étiquette et global.**

Système	Inférence	Neutre	Contradiction	Test
1	74,35%	68,13%	65,13%	69,27%
2	**86,10%**	**76,14%**	**77,70%**	**80,07%**
3	74,26%	69,18%	67,57%	70,39%

TABLE 2 – **Résultats des taux de réussite en pourcentage des étiquettes prédites pour le corpus de test de SNLI, par étiquette et global.**

où M est la longueur de la prémisse, N est la longueur de l'hypothèse, $\alpha \in \mathbb{R}^{M \times N}$ est un $softmax(e)$ sur la prémisse, et $\beta \in \mathbb{R}^{M \times N}$ est un $softmax(e)$ sur l'hypothèse. Ceci permet à la prémisse de voir le contexte de l'hypothèse et vice-versa.
Avec ces nouvelles représentations pour les mots, on effectue du *mean-pooling* :

$$pool_p = \frac{\sum_{i=1}^{N} p_i^c}{N}, pool_h = \frac{\sum_{i=1}^{M} h_i^c}{M} \tag{13}$$

puis on effectue une concaténation du *mean-pooling* de la prémisse et de l'hypothèse pour l'envoyer à une couche de décision :

$$y = softmax(W \times [pool_p; pool_h] + b) \tag{14}$$

5.1 Résultats des systèmes

Les tables 1 et 2 montrent les différents résultats des taux de réussite des étiquettes prédites sur le corpus SNLI [4] (plus de détail sur le corpus SNLI en section 7.1).

L'état de l'art pour le taux de réussite des étiquettes prédites est à 89,3% sur le corpus de test de SNLI. Nous avons cependant préféré avoir des systèmes rapide à implémenter avec des performances raisonnables. Nous appliquerons BAXI sur l'ensemble des systèmes publiquement disponibles dans un futur travail.

6 Description de l'approche BAXI

Cette section décrit notre technique pour interpreter une prédiction d'une entrée. L'objectif est de donner des explications pour chaque étiquette en donnant les α mots les plus importants dans la prémisse et les β mots les plus importants dans l'hypothèse. On parle alors "d'expliquer une étiquette". On veut donc calculer l'importance de chaque mot. L'importance d'un mot correspond à l'importance de sa contribution pour l'étiquette y.

Notre intuition est la suivante : On veut donner une explication pour l'étiquette y et l'entrée x composée de la prémisse et de l'hypothèse. On remplace alors un mot et on demande à notre modèle de nous donner les probabilités de chaque étiquette avec cette nouvelle entrée. Si la probabilité de l'étiquette y a baissé, alors le mot était important : cela veut dire que le mot avait contribué à l'étiquette y. A l'inverse, si elle a augmenté, le mot n'avait donc pas contribué à y. De plus, si la probabilité des autres étiquettes a augmenté, alors le mot a d'autant plus d'importance : en le remplaçant, l'entrée x a basculé vers une autre étiquette.

Pour résumé, lorsque l'on remplace m_i, nous pénalisons l'augmentation de la probabilité de l'étiquette que l'on veut expliquer, et nous encourageons l'augmentation des probabilités des autres étiquettes.
L'importance d'un mot est une fonction IMP : $\mathbb{R}^{d_e} \to \mathbb{R}$ comme suit :

$$\overset{y_{ref}}{\text{impact}}(m_i) = -p(y_{ref} \mid x \leftarrow m_i = m_a) \; + \; p(y_{ref} \mid x) \tag{15}$$

$$\overset{y_j}{\text{impact}}(m_i) = \sum_{j=1, j \neq ref}^{|Y|} p(y_j \mid x \leftarrow m_i = m_a) \; - \; p(y_j \mid x) \tag{16}$$

$$\text{IMP}(m_i) = \max_{m_a} (\overset{y_{ref}}{\text{impact}}(m_i) \; + \; \overset{y_j}{\text{impact}}(m_i)) \tag{17}$$

où y_{ref} est l'étiquette de référence (l'étiquette que l'on veut "expliquer"), x est l'entrée, m_i est le mot que l'on retire de l'entrée x, m_a est le mot par lequel on remplace m_i, y_j est une étiquette différente de y_{ref}, et $|Y|$ est le nombre d'étiquettes.

L'équation 9 pénalise l'augmentation de la probabilité de l'étiquette que l'on veut expliquer lorsque l'on remplace m_i par m_a.
L'équation 10 encourage l'augmentation des probabilités des autres étiquettes lorsque l'on remplace m_i par m_a.
Pour calculer l'importance d'un mot, on additionne ces deux impacts. Cette formule suit notre intuition de base et est similaire à celle proposée par Robnik-Sikonja et Kononenko [15], mais également à celle proposée par Fong et Vedaldi [5] décrit en section 3.2. Nous avons rajouté la prise en compte de l'impact sur les probabilités des autres étiquettes.
Nous cherchons ici le mot de remplacement m_a qui maximise cette mesure d'importance, pour trouver le meilleur exemple adversarial pour l'entrée x. Ce que l'on cherche à faire est donc de comparer l'entrée x avec le meilleur exemple adversarial, caractérisé par le remplacement de m_i par m_a.

Cette méthode permet donc d'avoir une explication contrastée, puisque l'on

compare notre entrée de base avec des entrées créées artificiellement en remplacant un mot par un autre. De plus, elle est également sélective puisque l'on sélectionne les α mots dans la prémisse et les β mots dans l'hypothèse ayant le plus contribué à l'étiquette y_{ref}. Enfin, le programme surligne les mots les plus importants pour chaque étiquette, ce qui permet d'avoir une visualisation pour faciliter la compréhension. Nous respectons donc les trois règles d'une bonne explication citées dans la section 4.2.

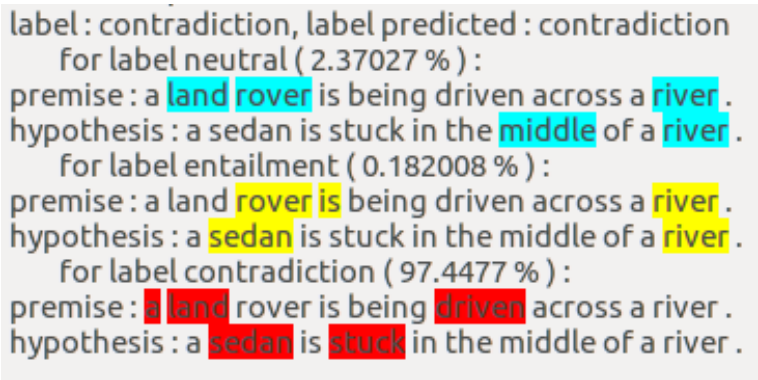

FIGURE 3 – **Exemple d'une explication fournie par BAXI sur le système 2. Les mots surlignés en bleu ont contribué à la neutralité, les mots surlignés en jaune ont contribué à l'inférence, et les mots surlignés en rouge ont contribué à la contradiction. La probabilité de chaque étiquette est spécifiée entre parenthèse.**

7 Cadre expérimental

7.1 Corpus SNLI et représentation de mots

Nous utilisons les corpus SNLI annotés pour la tâche de la RTE, composés d'un fichier d'entraînement, de validation et de test, contenant respectivement 550 152, 10 000 et 10 000 exemples composés de paires prémisse/hypothèse avec l'étiquette correspondante.
Ces corpus ont été réalisé par cinq annotateurs à l'aide d'une image accompagnée d'un texte bref -la prémisse- présentant la dite image. Les annotateurs ont alors écrit une phrase étant neutre par rapport à la prémisse, une autre étant en contradiction et une autre phrase qui pouvait être déduite de la prémisse : ils ont donné une hypothèse et une étiquette. Pour que le corpus ne soit pas trop subjectif, les annotateurs ont eu accès à quelques paires prémisse/hypothèse sans étiquette. Chacun d'entre eux a donné une étiquette, celle ayant eu le plus de voix a été décidé comme l'étiquette *gold* de la paire observée. Ainsi, certaines paires n'ont pas d'étiquette car les annotateurs n'ont pas trouvé de consensus : nous ne prennons pas en compte ce genre d'entrée. La table 3 est un échantillon du corpus SNLI.
Pour la représentation des mots, nous utilisons des *words embeddings* pré-entrainés via GloVe.6B.100d. Pour les mots inconnus, c'est-à-dire les mots qui n'ont pas d'*embedding* dans GloVe, nous utilisons des *embeddings* initialisés au hasard.

A man inspects the uniform of a figure in some East Asian country.	**contradiction** C C C C C	The man is sleeping
An older and younger man smiling.	**neutral** N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	**contradiction** C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	**entailment** E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	**neutral** N N E C N	A happy woman in a fairy costume holds an umbrella.

TABLE 3 – **Echantillon de 5 paires du corpus de développement de SNLI présentant à gauche la prémisse, à droite l'hypothèse, et au centre les étiquettes des 5 annotateurs (C pour Contradiction, N pour *Neutral* (neutre), et E pour *Entailment* (inférence)) avec en premier l'étiquette de l'auteur de la paire. L'étiquette en gras est celle qui a eu le plus de voix, et est donc l'étiquette *gold* de la paire prémisse/hypothèse.**

7.2 Corpus explicatif

La méthode habituelle pour ce genre d'expériences est de montrer à plusieurs personnes les paires de phrases prémisse/hypothèse avec leurs explications associées pour connaître leur avis sur la qualité de l'explication.
Nous proposons une autre méthode qui permet d'évaluer automatiquement la qualité d'une explication : nous avons montré un échantillon de 43 exemples issus du fichier de test de SNLI à six personnes pour qu'elles puissent annoter ce qu'elles pensent être une explication correcte, c'est-à-dire les mots de la prémisse et les mots de l'hypothèse qui conduisent à l'étiquette associée. Nous avons récolté leurs réponses pour en faire un corpus d'explications de références, au format CSV. La figure 4 est un échantillon de ce corpus explicatif.

7.3 Corpus d'exemples adversariaux

Pour que BAXI marche bien, il faut remplacer les mots par d'autres mots pertinents pour créer des exemples adversariaux. Pour cela, on a annoté trois fichiers -un pour chaque étiquette- contenant 19 paires se trouvant dans le corpus explicatif. Nous n'avons pas encore pu annoter les 43 exemples (la totalité du corpus explicatif), ce sera fait dans un futur travail. Dans ces fichiers, on spécifie pour chaque mot de chaque paires, les mots pouvant le remplacer.

L'idée pour l'annotation des mots de remplacement est la suivante : si l'étiquette que l'on veut expliquer se trouve être l'étiquette *gold* de la paire prémisse/hypothèse, on cherche alors à remplacer les mots des phrases par des mots pouvant baisser fortement la probabilité de l'étiquette. Par exemple, si l'étiquette *gold* est inférence, on va chercher à basculer vers la neutralité ou la contradiction pour faire baisser l'inférence et ainsi créer un exemple adversarial.

étiquette	Prémisse	Hypothèse	Mots explicatifs (prémisse)	Mots explicatifs (hypothèse)
inférence	This church choir sings to the masses as they sing joyous songs from the book at a church.	The church is filled with song.	sings 3 songs 11 church 17	filled 3 song 5 church 1
contradiction	A man playing an electric guitar on stage.	A man playing banjo on the floor.	man 1 guitar 5 stage 7	man 1 banjo 3 floor 6
neutre	An old man with a package poses in front of an advertisement.	A man poses in front of an ad for beer.		for 8 beer 9
inférence	A blond-haired doctor and her African american assistant looking threw new medical manuals.	A doctor is looking at a book	doctor 2 looking 8 manuals 12	doctor 1 looking 3 book 6

TABLE 4 – **Echantillon du corpus explicatif. Chaque ligne correspond à une paire. Le nombre à droite de chaque mot explicatif est sa position dans sa phrase correspondante : certains mots pouvent être présent plusieurs fois dans la phrase, tel que "church" dans la prémisse du premier exemple. Il faut donc les différencier. Le troisième exemple n'a pas de mot explicatif dans la prémisse.**

A l'inverse, si l'étiquette que l'on veut expliquer n'est pas l'étiquette *gold*, on cherche alors à remplacer les mots de la phrase par des mots pouvant augmenter la probabilité de l'étiquette expliquée (et par la même occasion, faire baisser la probabilité des autres étiquettes). Par exemple, si l'étiquette *gold* est neutre et que l'on veut expliquer la contradiction, on cherche à créer une contradiction avec les mots de remplacement. Les tables 7.3 et 7.3 présentent un exemple pour la prémisse et l'hypothèse.

La table 7.3 présente la distribution des étiquettes présentes dans les 19 exemples que nous évaluons.

7.4 Paramètres

Tous les *embeddings* sont mis-à-jour par le réseau. Ils sont de dimension 100. La taille des *batches* est de 16.
Concernant les RNNs utilisés, il n'y a qu'une seule couche, la dimension des états cachés est de 100, et le *dropout* est à 0,3.
L'entraînement des systèmes 1, 2 et 3 est réalisé avec l'optimiseur Adam et *Negative softmax log likelihood* est utilisé pour calculer le *loss* pendant 20 époques. Pour l'utilisation de LIME, on a fixé le nombre de voisin de l'entrée à 500.

8 Résultats

Pour contextualiser, la table 8 montre les taux de réussite des étiquettes prédites pour les trois systèmes décrits en section 5 sur le petit corpus d'évaluation

Prémisse	Liste de mots remplaçants
A	Another That This The
woman	man boy person guy dude
with	without and wearing selling
a	that the this those these
green	blue sad happy depressed
headscarf	happiness joy sadness depression
,	and also ;
blue	sad depressed red green
shirt	headscarf jacket glasses scarf shoes
and	also , ;
a	that the this those these
very	small depressed sad few
big	young old small depressed sad
grin	sandwich headscarf shirt depression sadness
.	!

Hypothèse	Liste de mots remplaçants
The	Another A That This
woman	man boy person guy dude
is	was were
very	little few not
happy	old young tall small depressed
.	!

TABLE 5 – **Exemple d'une prémisse (table d'en haut) et d'une hypothèse (table d'en bas) avec une liste de mots pouvant remplacer le mot correspondant. L'étiquette de cette paire est inférence, et on veut expliquer l'inférence. Il faut donc des mots de remplacement qui fassent baisser l'inférence.**

contenant 19 exemples.

La table 8 illustre les différents résultats obtenus pour le taux de réussite des explications des trois systèmes décrits en section 5 sur le petit corpus d'évaluation, à l'aide du corpus explicatif. Le taux de réussite des explications correctes est mesuré par le nombre de mots explicatifs correctement trouvés sur le nombre de mots explicatifs total pour le petit corpus d'évaluation. Les vrais mots explicatifs se trouvent dans le corpus explicatif.

/* mettre des graphiques */

9 Discussion

BAXI a cependant des limites que nous allons corriger dans de futurs travaux : en effet, BAXI remplace seulement un seul mot dans l'entrée, ce qui est problématique face aux expressions polylexicales. Par exemple, si l'entrée contient l'expression *"in front of"*, BAXI remplace le mot *"in"* pour faire une

Inférence	Neutre	Contradiction
36,84%	26,31%	36,84%

TABLE 6 – **Distribution en pourcentage des étiquettes présentes dans le petit corpus d'évaluation contenant 19 exemples issus du corpus de test de SNLI.**

Système	% Inférence	% Neutre	% Contradiction	% Total
1	**71,43%**	60,00%	42,86%	**57,89%**
2	42,86%	**80,00%**	**57,14%**	**57,89%**
3	**71,43%**	40,00%	42,86%	52,63%

TABLE 7 – **Résultats des taux de réussite des étiquettes prédites sur le CPI.**

Système	Méthode	% Inférence	% Neutre	% Contradiction	% Total
1	BAXI	**61,70%**	**57,14%**	**51,61%**	**57,61%**
	LIME	27,66%	28,57%	32,26%	29,35%
2	BAXI	**76,60%**	**57,14%**	54,84%	**66,30%**
	LIME	55,32%	50,00%	**61,29%**	56,52%
3	BAXI	**61,70%**	**57,14%**	51,61%	**57,61%**
	LIME	31,91%	35,71%	**61,29%**	42,39%

TABLE 8 – **Résultats des mesures de BAXI et LIME sur le CPI avec le corpus explicatif.**

nouvelle entrée, puis le mot *"front"*, et enfin le mot *"of"*, car un seul mot peut être remplacé. Cela ne donne pas forcément de nouvelles entrées ayant du sens. BAXI est donc actuellement en ordre 1, l'idéal serait qu'il soit en ordre N pour remplacer N mots.
Une autre limite de la méthode est l'explication de la neutralité, car il est parfois difficile de l'expliquer avec des mots contribuant à la neutralité. Peut-être faudrait-il expliquer ce qui est contre l'inférence et contre la contradiction dans une entrée étiquettée "neutre".
Par ailleurs, nous avons noté à la main les mots pouvant remplacer chaque mot dans l'entrée. Cela peut se faire de manière automatique à l'aide d'outils tel que WordNet, cependant il faudrait alors reconnaitre les expressions polylexicales, de plus cela deviendrait un exercice aussi difficile que l'inférence textuelle. Nous pensons qu'il est préférable de les annoter à la main.
Concernant SNLI, le corpus contient des biais que BAXI arrive à retrouver, surtout concernant la neutralité : les annotateurs ont souvent pris l'idée que pour qu'une entrée soit neutre, l'hypothèse doit rajouter un but [16]. Ainsi, BAXI surligne le mot *"for"* en faveur de la neutralité quand il le voit dans l'hypothèse. Il faudrait cependant avoir plus de données de test pour être certain qu'il retrouve toujours ce genre de biais, ce qui est dans nos projets futurs.
Ces biais sont la conséquence de la façon dont SNLI a été construit (voir section 7.1 pour l'explication de la construction de SNLI). De plus, il arrive que certains éléments de l'image présentée ne se trouvent pas dans la prémisse, ce qui engendre des étiquettes difficiles à prédire. La figure 4 en est un exemple.

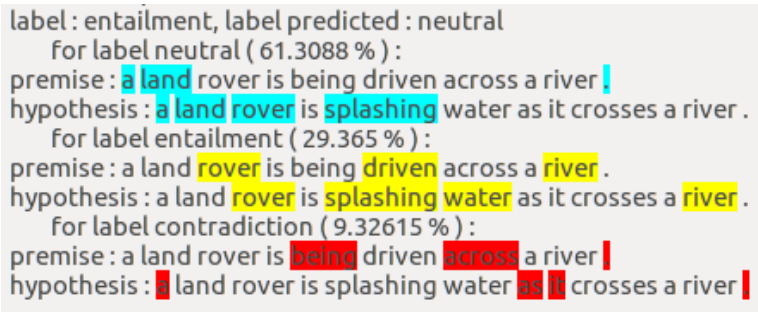

FIGURE 4 – **Exemple d'une explication fournie par BAXI sur le système 2. Les mots surlignés en bleu ont contribué à la neutralité, les mots surlignés en jaune ont contribué à l'inférence, et les mots surlignés en rouge ont contribué à la contradiction. L'étiquette *gold* est "inférence", cependant il n'y a aucune mention d'éclaboussures d'eau dans la prémisse. Selon BAXI, *"splashing"* a donc contribué à la neutralité. Il a également contribué à l'inférence, ce qui n'est pas surprenant puisqu'il est en présence de mots tels que *"river"* et *"water"*.**

En outre, lorsque les annotateurs choisissent une hypothèse et une étiquette, il faudrait qu'ils annotent aussi les mots ayant conduits à leur choix pour pouvoir directement avoir un corpus explicatif pour mesurer le taux d'explications correctement donnés. Nous pensons qu'en plus du taux de réussite des étiquettes prédites, il faut prendre en compte le taux de réussite des explications pour

savoir si un modèle est bon ou non.

10 Conclusion

Pour conclure, nous avons présenté la méthode BAXI pour interpréter les prédictions de modèles dans le cadre de la RTE. BAXI donne les mots dans la prémisse et dans l'hypothèse ayant contribués à l'étiquette que l'on veut "expliquer", en donnant une valeur d'importance à chaque mot. Pour cela, BAXI remplace un seul mot dans une entrée. Le mot retiré est remplacé par un mot de sorte à créer un exemple adversarial. On regarde alors l'impact de ce changement sur les probabilités de l'étiquette pour pouvoir donner une valeur d'importance au mot retiré : si la probabilité de l'étiquette que l'on veut expliquer a augmenté, alors le mot n'avait pas d'importance pour l'étiquette considérée. De plus, si la probabilité des autres étiquettes a baissé, alors le mot n'avait pas d'importance non plus. Nous avons comparé BAXI à la méthode LIME qui explique n'importe quel prédiction de modèle en apprenant localement un modèle interprétable dans le voisinage de l'entrée. Contrairement à BAXI, LIME créé des nouvelles entrées qui n'ont pas forcément de sens pour créer un voisinage, car LIME retire un ou plusieurs mots sans les remplacer.
Pour se faire, nous avons créé un corpus explicatif comprenant, pour quelques entrées tirées du corpus de test de SNLI, les mots expliquant l'étiquette associée à l'entrée. Nous comptons alors le taux de bons mots explicatifs trouvés par les deux méthodes, uniquement pour les étiquettes *gold*. Les résultats montrent que BAXI trouve de meilleurs explications. Pour nos futurs travaux, nous améliorerons BAXI en lui permettant de remplacer plusieurs mots au lieu d'un seul, ce qui résoudrait le problème des expressions polylexicales. Nous allons également réaliser plus d'exemples pour tester la qualité des explications.

11 Remerciements

J'adresse mes sincères remerciements à M. Benoit Favre pour ses conseils, son écoute, et la confiance qu'il m'a accordée durant ce projet, ce qui m'a permit de m'accomplir totalement dans mes missions.
Je remercie également l'ensemble de l'équipe TALEP pour leur accueil et leur aide lors des différentes réunions, ainsi que l'ensemble des membres du jury pour avoir accepté la charge de la lecture critique et de l'évaluation de ce travail.
Enfin, je remercie les annotateurs du corpus explicatif, Thomas Delfino, Franck Dary, Simone Fuscone, Sébastien Ratel, et Jérémy Auguste pour m'avoir accordé un peu de temps lors de la création de ce corpus.

Références

[1] Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin. 'why should i trust you ?' : Explaining the predictions of any classifier. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

[2] Oren Glickman, Ido Dagan, and Moshe Koppel. Web based probabilistic textual entailment. In *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*, 2005.

[3] Bill MacCartney and Christopher D Manning. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 2007.

[4] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[5] Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *arXiv preprint arXiv :1704.03296*, 2017.

[6] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *arXiv preprint arXiv :1704.02685*, 2017.

[7] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederik Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. In *PloS One*, 2015.

[8] Scott Lundberg and Lee Su-In. A unified approach to interpreting model predictions. In *NIPS*, 2017.

[9] Lloyd S. Shapley. A value for n-person games. In *Contributions to the Theory of Games*, 1953.

[10] Christoph Molnar. *Interpretable Machine Learning*. Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, 2018.

[11] Vivian S. Silva, André Freitas, and Siegfried Handschuh. Building a knowledge graph from natural language definitions for interpretable text entailment recognition. In *LREC*, 2018.

[12] Tim Miller. Explanation in artificial intelligence : Insights from the social sciences. *arXiv preprint arXiv :1706.07269*, 2017.

[13] Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. Dynet : The dynamic neural network toolkit. *arXiv preprint arXiv :1701.03980*, 2017.

[14] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. Natural language inference with external knowledge. *arXiv preprint arXiv :1711.04289*, 2017.

[15] Marko Robnik-Sikonja and Igor Kononenko. Explaining classifications for individual instances. In *IEEE Transactions on Knowledge and Data Engineering*, 2008.

[16] Suchin Gururangan, Swabha Swayamdipta, Roy Schwartz, Omer Levy, Samuel R. Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *NAACL*, 2018.