Etat de l'art : Recherche d'incohérences dans un texte

Avant de commencer notre implémentation d'un modèle NLI (Natural Language Inference), nous avons fait des recherches sur les modèles déjà existant sur le marché. Lors de cette recherche, nous avons utiliser principalement le site GLUE qui est le référentiel d'évaluation de la compréhension générale du langage. C'est une collection d'ensembles de données utilisés pour la formation, l'évaluation et l'analyse des modèles de NLP les uns par rapport aux autres, dans le but de conduire "la recherche dans le développement de systèmes généraux et robustes de compréhension du langage naturel". La collection se compose de neuf ensembles de données de tâches «difficiles et diverses» conçues pour tester la compréhension du langage d'un modèle, et est essentielle pour comprendre comment les modèles d'apprentissage par transfert comme le BERT sont évalués.

Grâce à ce site qui possède un classement des meilleurs modèles en fonction de leurs scores, nous avons pu piocher quelques modèles qui font partie du classement afin de savoir comment ils fonctionnent. Nous avons donc utilisé, pour notre état de l'art, les modèles suivants :

- BiLSTM + Attn
- CAFE
- GenSen
- RoBERTa
- Snorkel Metal
- Finetuned Transformer LM
- MT-DNN-ensemble
- Hierarchical BiLSTM Max Pooling
- XLNet-Large

Nous avons résumé, pour chacun d'eux, leurs positions dans le classement GLUE ainsi que le domaine où ils sont le plus efficaces. Nous avons aussi proposé un petit résumé des architectures des modèles ainsi que comment ils fonctionnent. Pour avoir plus d'informations sur chacun des modèles présentés ci-dessous, je vous invite à regarder le GIT où se trouve un résumé rapide et efficace, fait par nous même, de chaque modèle. Certains modèle sont spécialisés dans des tâches précises donc ils n'obtiennent pas forcément un bon score général mais ils peuvent être utiles dans notre cas.

BiLSTM + Attn

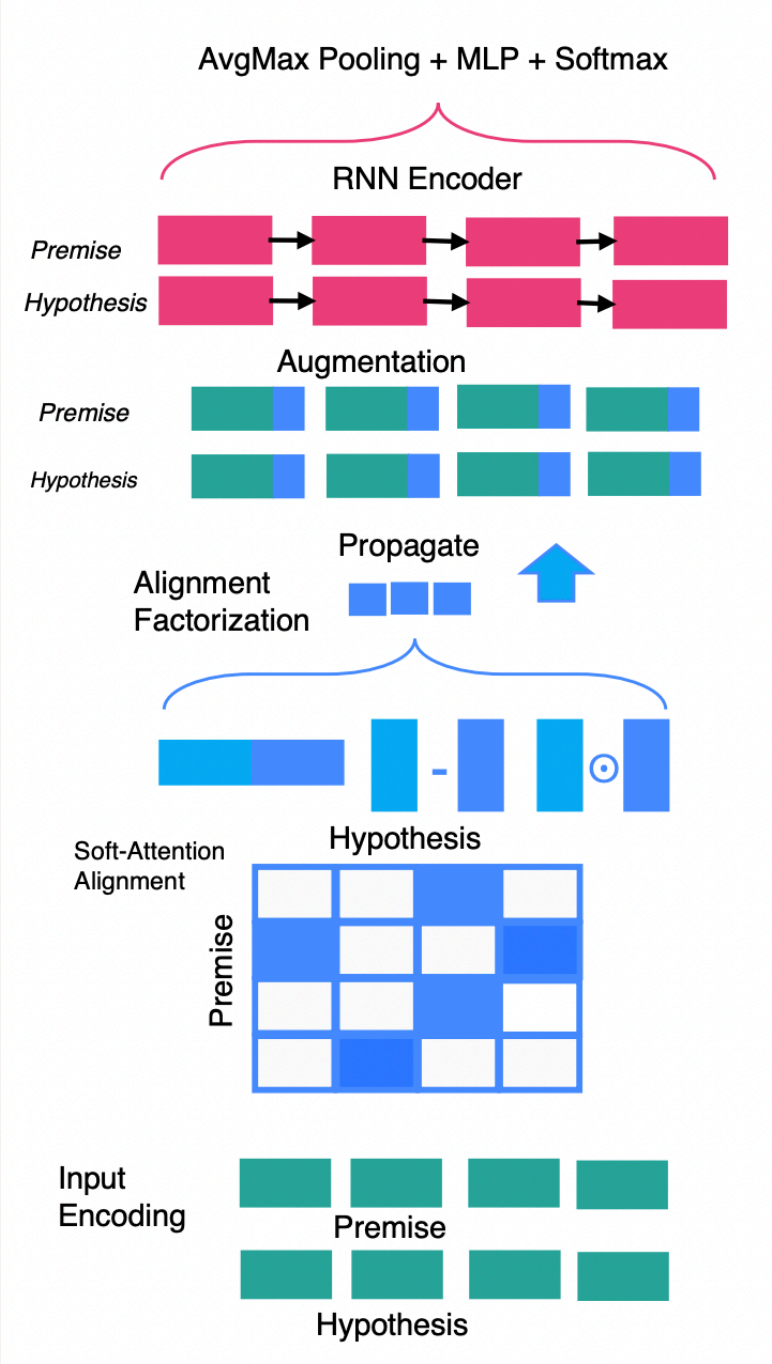
Ce modèle ressemble très fortement au modèle Multi-task BiLSTM + Attn en ce qui concerne l'architecture.

Le modèle utilise un BiLSTM (LSTM bidirectionnels) à deux couches, 1500D avec max pooling et 300D GloVe word embeddings. Pour les tâches à une seule phrase, nous encodons la phrase et passons le vecteur résultant à un classificateur. Pour les tâches de paires de phrases, nous encodons les phrases indépendamment pour produire des vecteurs u, v, et passons [u , v ; |u - v| ; u ∗ v] à un classificateur (détail de cette phase dans un autre état de l'art). Le classifieur est un MLP avec une couche cachée de 512D.

CAFE

CAFE possède une architecture d'apprentissage profond pour l'inférence en langage naturel (NLI). L'objectif principal du modèle est, étant donné une prémisse et une hypothèse, de détecter si la seconde implique ou contredit la première.

L' architecture du modèle prend en entrée une prémisse et une hypothèse qui sont ainsi comparées, compressées puis propagées vers des couches supérieures pour obtenir un apprentissage amélioré de la représentation. Ensuite, des couches de factorisation sont adaptées pour une compression efficace et expressive des vecteurs d'alignement en caractéristiques scalaires, qui sont ensuite utilisées pour augmenter les représentations des mots de base.



GenSen

GenSen est une technique pour apprendre des objectifs, des représentations de phrases en utilisant un apprentissage multitâche.

Concernant la préparation de l'entraînement multitâches :

Nous avons besoin d'un certain nombre de tâches, d'un encodeur partagé pour toutes les tâches et un nombre de décodeurs (du même nombre que celui des tâches). Maintenant, les prochaines étapes vont boucler : Nous récupérons une tâche i et son jeu de données associé pour créer des paires de valeurs entrée/sortie. La valeur d'entrée va alors être encodée pour créer une représentation qui va ensuite passer dans le décodeur associé à la tâche, pour générer une prédiction.

Le modèle utilisé est comme suit :

L'encodeur partagé utilise une table de recherche de mots communs et un GRU (Gated Recurrent Unit) qui est en quelque sorte un réseau de neurones artificiel qui peut non seulement utiliser des données simples mais également des séquences de données entières comme de la vidéo.

RoBERTa

BERT est un modèle de langage, un modèle qui modélise la distribution de séquences de mots ou de lettres. Il a par exemple été utilisé dans la prédiction d'un mot en fonction des mots qui le précédent.

RoBERTa est une méthode d'apprentissage basée sur BERT mais améliorée, qui atteint des résultats similaires voire supérieurs à bon nombre de méthodes. Ces améliorations consistent notamment en l'agrandissement des données d'entraînement (et donc en entraînant le modèle plus longtemps), en enlevant l'objectif initial qui était la prédiction des mots suivants, en l'entraînant sur des phrases plus longues, et en changeant dynamiquement les motifs (les patterns) des données d'entraînement.

BERT prend en entrée 2 segments (chaque segment constitué de plus d'une phrase) séparés par des tokens spécifiques : Les segments étant x1,…,xn et y1,…,ym : [CLS],x1,…,xn,[SEP],y1,…,ym,[EOS] BERT aura alors 2 objectifs pendant l'entraînement :

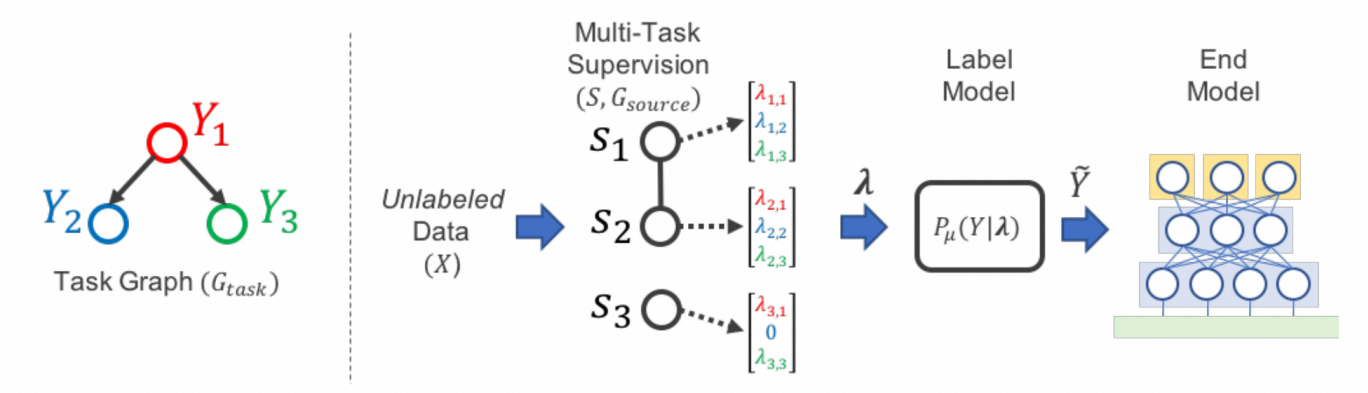
- Masked Language Model (MLM) qui consiste à remplacer aléatoirement des tokens dans la donnée d'entrée par le token [MASK] afin d'ensuite prédire quel était le token d'origine.
- Next Sentence Prediction (NSP) qui consiste à prédire si deux segments se suivent ou non dans le texte d'origine.

Snorkel Metal :

MeTal est un framework utilisé pour la modélisation et l'intégration de sources de supervision faible avec des exactitudes, des corrélations et des granularités inconnues.

Voici comment MeTal fonctionne :

L'utilisateur donne en entrée un graphe de tâches labellisées (Y) (montrant les liens entre chaque tâche), un ensemble de points de données non labellisés (X), un ensemble de sources de supervision faible multitâches (S) (qui retournent tous un vecteur de labels de tâches pour l'ensemble de données précédent) et un deuxième graphe : la structure de dépendance entre ces sources.



L'architecture consiste en :

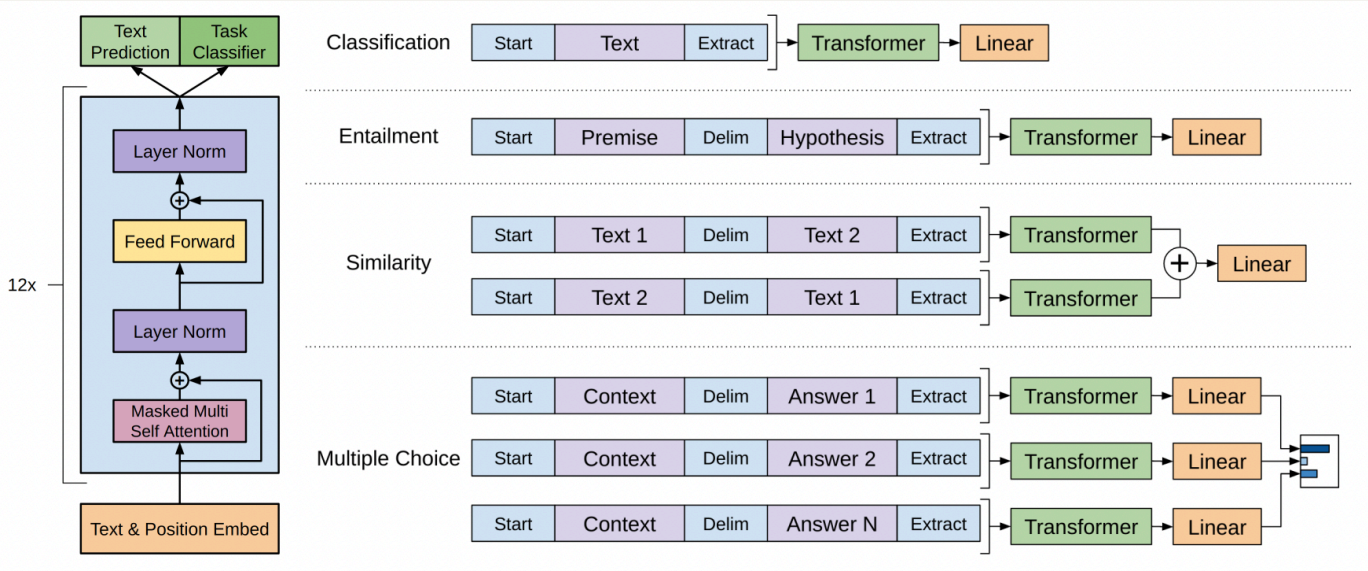
- Une couche d'entrée LSTM (Long Short-Term Memory, réseau de neurones artificiel pouvant utiliser des données simples comme des séquences de données tels des vidéos) bidirectionnelle partagée avec des incorporations pré-entraînées.
- Des couches intermédiaires linéaires partagées.
- Une couche finale linéaire pour chaque tâche.

Finetuned Transformer LM

Un Language Model (LM) est un modèle qui apprend à prédire le prochain mot d'une phrase en fonction de sa connaissance des mots précédents. En faisant cela, le LM apprend à comprendre la langue du corpus d'entraînement.

Son architecture:

- C’est un modèle deep learning comme par exemple RNN.
- Encoder : à partir d’une phrase en entrée, il va produire un vecteur d’activation qui représente sa compréhension du modèle.
- Classifier : il s’agit de l’ensemble des dernières couches d’un LM qui à partir du vecteur d’activations précédent va prédire un mot sous la forme d’un vecteur d’embeddings.

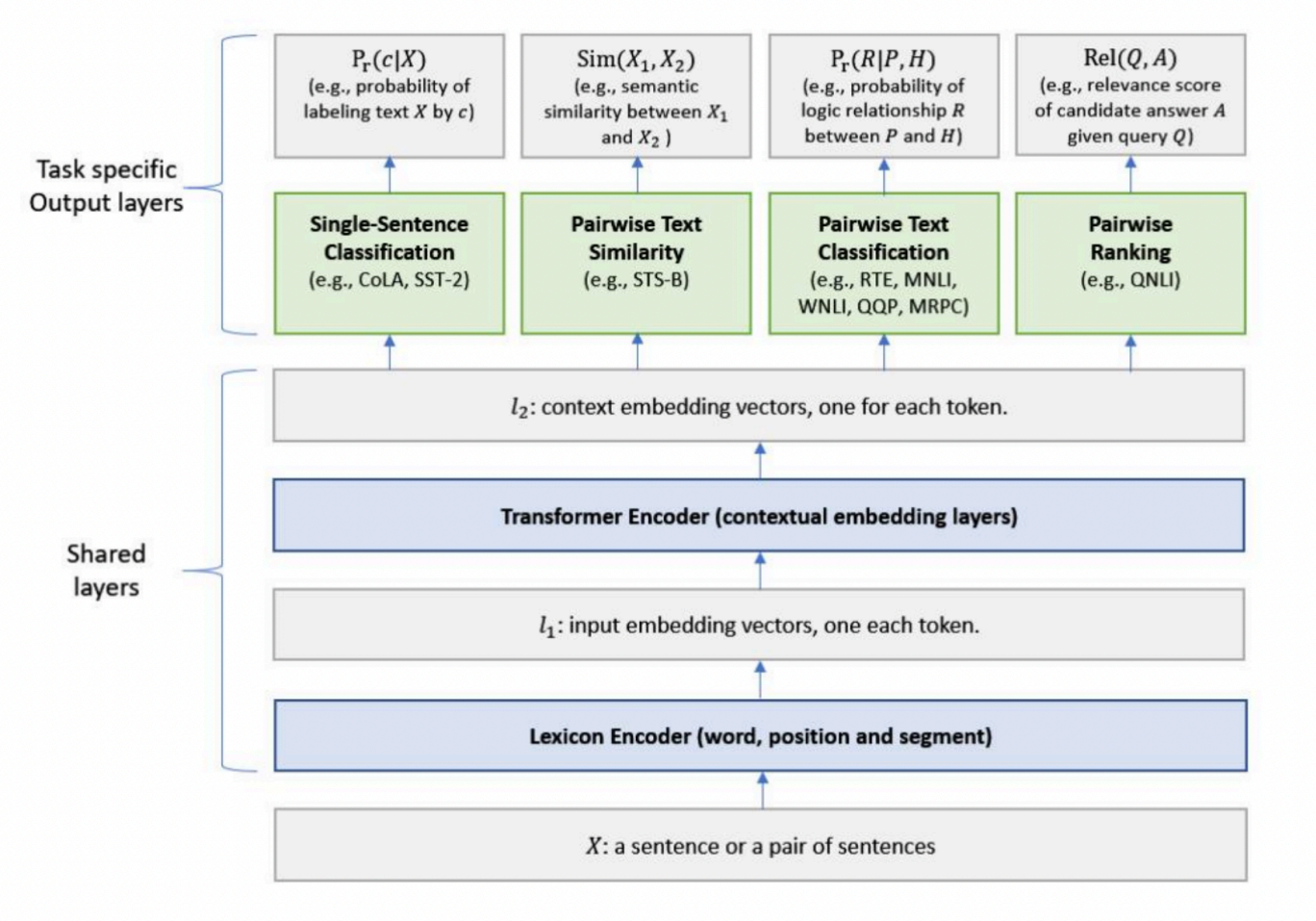


A droite de l'image, les transformations d'entrée pour affiner différentes tâches. Le modèle convertit toutes les entrées structurées en séquences de jetons à traiter par le modèle pré-formé, suivi d'une couche linéaire + softmax.

MT-DNN-ensemble

MT-DNN s'appuie sur un modèle proposé par Microsoft en 2015 et intègre l'architecture réseau de BERT, un modèle de langage de transformateur bidirectionnel pré-entraîné proposé par Google l'année dernière.

L'implémentation de MT-DNN est basée sur les implémentations PyTorch de MT-DNN3 et BERT4 .De plus, MT-DNN utilise Adamax comme optimiseur.



Comme le montre la figure ci-dessus, les couches de bas niveau du réseau (c'est-à-dire les couches d'encodage de texte) sont partagées entre toutes les tâches, tandis que les couches supérieures sont spécifiques aux tâches, combinant différents types de tâches NLU (Natural language understanding). Comme le modèle BERT, MT-DNN est formé en deux phases : pré-formation et mise au point. Mais contrairement à BERT, MT-DNN ajoute l'apprentissage multitâche (MTL) dans les phases de réglage fin avec plusieurs couches spécifiques aux tâches dans son architecture de modèle.

L'entrée X, qui est une séquence de mots (soit une phrase soit un ensemble de phrases regroupées) est d'abord représentée comme une séquence de vecteurs d'intégration, un pour chaque mot, dans *l1*. Ensuite, le codeur transformateur capture les informations contextuelles pour chaque mot et génère les vecteurs d'intégration contextuelle partagés dans *l2*. C'est la représentation sémantique partagée qui est entraînée par nos objectifs multitâches.

Enfin, pour chaque tâche, des couches supplémentaires spécifiques à la tâche génèrent des représentations spécifiques à la tâche, suivies des opérations nécessaires à la classification, à la notation de similarité ou au classement de pertinence.

Hierarchical BiLSTM Max Pooling

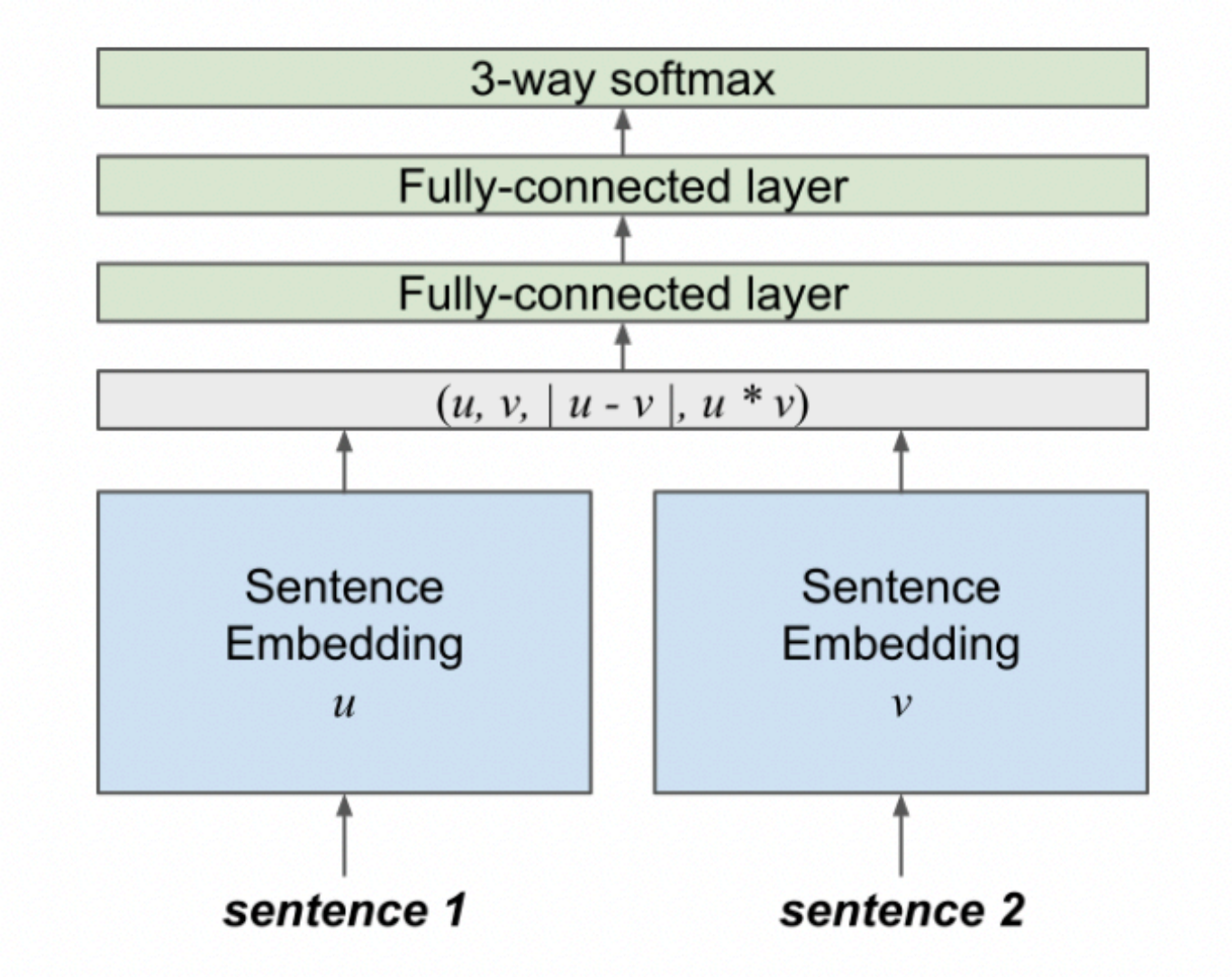
L' architecture proposée suit une approche basée sur l'intégration de phrases pour NLI.

Le modèle propose une hiérarchie de couches BiLSTM (LSTM bidirectionnels) et max pooling qui implémente une stratégie de raffinement itérative et produit des résultats de pointe sur l'ensemble de données SciTail ainsi que des résultats solides pour SNLI et MultiNLI.

Les BiLSTM utilisent deux LSTM pour s’entraîner sur l’entrée séquentielle.

Les cellules LSTM (Long Short Term Memory) possèdent une mémoire interne appelée cellule. La cellule permet de maintenir un état aussi longtemps que nécessaire. Cette cellule consiste en une valeur numérique que le réseau peut piloter en fonction des situations.

L'idée de base derrière ces approches est d'encoder séparément les phrases de prémisse et d'hypothèse, puis de les combiner à l'aide d'un classificateur de réseau neuronal.



Le modèle illustré sur l'image ci-dessus contient des incorporations de phrases pour les deux phrases d'entrée, où la sortie des incorporations de phrases est combinée à l'aide d'une heuristique introduite par Mou, en rassemblant la concaténation (u, v), la différence absolue par élément |u - v|, et le produit par élément u * v.

Le vecteur combiné est ensuite transmis à un perceptron multicouche à 3 couches (MLP) avec un classificateur softmax à 3 voies.

Les deux premières couches du MLP utilisent à la fois l'abandon et une fonction d'activation ReLU.

XLNet-Large

XLNet est un modèle autorégressif (AR) généralisé pour la compréhension de langage naturel où chaque mot dépend de tous les mots précédents. Il est généralisé parce qu'il capture le contexte de façon bidirectionnelle au moyen d'un mécanisme de modélisation par permutation de langage (‘permutation language modeling’ (PLM)) en entrainant un modèle AR sur toutes les permutations possibles de mots dans une phrase tout en conservant l'ordre des mots dans la séquence originale.

Les modèles XLNet de base et large ont les mêmes hyper paramètres d'architecture que BERT, ce qui donne des modèles de même grandeur.

XLNET est un modèle où le jeton suivant dépend de tous les jetons précédents. XLNET est "généralisé" car il capture le contexte bidirectionnel au moyen d'un mécanisme appelé "modélisation du langage de permutation". Il intègre l'idée de modèles auto-régressifs et de modélisation de contexte bidirectionnelle, tout en surmontant les inconvénients de BERT.

Pour implémenter XLNET, le transformateur est modifié pour ne regarder que la représentation cachée des jetons précédant le jeton à prédire.

Résultats obtenus :

Les scores obtenus pour les différents modèles proviennent tous de la plateforme GLUE, pour avoir ainsi les mêmes tests effectués sur chaque modèle et donc ainsi avoir des résultats comparables. Voici les résultats obtenus :

- RoBERTa : 88.1
- MT-DNN-ensemble : 87.6
- Snorkel Metal : 83.2
- Hierarchical BiLSTM Max Pooling : 70.0
- GenSen : 66.1
- BiLSTM + Attn : 65.6
- CAFE : (non classé car trop spécialisé)
- Finetuned Transformer LM : (non classé car trop spécialisé)
- XLNet-Large : (non classé car manque une tâche où il n'y a pas de score sinon moyenne d'environ 85.5)

On peut voir que les modèles RoBERTa, MT-DNN-ensemble et Snorkel Metal obtiennent de bons résultats d'ensemble. Ce sont des modèles potentiellement utilisable.

De plus, le but de notre projet est d'avoir un modèle solide dans les méthodes d'inférence à partir de langage naturel. Notre modèle doit donc obtenir de bons résultats à l'aide du corpus MultiNLI.

MultiNLI : Le corpus Multi-Genre Natural Language Inference (MultiNLI) est un corpus à large couverture pour l'inférence en langage naturel, composé de 433 000 paires de phrases écrites par l'homme étiquetées avec implication, contradiction et neutre. Contrairement au corpus SNLI, qui tire la phrase de prémisse des légendes d'images, MultiNLI se compose de paires de phrases de dix genres distincts d'anglais écrit et parlé. L'ensemble de données est divisé en ensembles d'entraînement (392 702 paires), de développement (20 000 paires) et de test (20 000 paires).

Voici donc un classement obtenus à l'aide de ce corpus sur un jeu d'entraînement et d'évaluation :

Model	Matched	Mismatched	Paper / Source	Code
RoBERTa (Liu et al., 2019)	90.8	90.2	RoBERTa: A Robustly Optimized BERT Pretraining Approach	Official
XLNet-Large (ensemble) (Yang et al., 2019)	90.2	89.8	XLNet: Generalized Autoregressive Pretraining for Language Understanding	Official
MT-DNN-ensemble (Liu et al., 2019)	87.9	87.4	Improving Multi-Task Deep Neural Networks via Knowledge Distillation for Natural Language Understanding	Official

Les scores ci-dessus proviennent du site NLP-progress qui est un référentiel pour suivre les progrès du traitement du langage naturel (NLP), y compris les ensembles de données et l'état de l'art actuel pour les tâches NLP les plus courantes. Ce site nous a été fourni par nos professeurs encadrant donc nous pensons pouvoir le prendre au sérieux.

On peut voir que RoBERTa obtient encore une fois les meilleurs scores. Nous allons utiliser ce modèle afin de constituer notre application d'analyse de texte, d'annotation du texte et d'identifications et d'élimination de contradictions/redondances.