Projet d'analyse d'un corpus structuré

Ben Gao[1] and Loïck Chardon[1]

[1]Master MIASHS, Université Lumière Lyon 2.

1 Introduction

Ce rapport présente le travail réalisé dans le cadre du projet d'analyse d'un corpus issu de la plateforme *Persée*. Unité d'appui et de recherche (UAR) rattachée à l'ENS de Lyon et au CNRS, Persée ouvre son portail en 2005. Sa mission est de valoriser le patrimoine documentaire de la recherche scientifique en numérisant et en diffusant une large collection de publication scientifique en lien principalement avec les sciences humaines et sociales. Il s'agit d'un outil de qualité pour rechercher, consulter et manipuler des publications scientifiques et autres métadonnées associées à ces vastes corpus numériques. Ainsi l'objectif principal de ce projet est de développer une solution permettant d'extraire des informations pertinentes à partir de cet ensemble de données textuelles, en mettant en œuvre différentes fonctionnalités telles que le chargement des données, la visualisation, le moteur de recherche, le clustering et la classification supervisée. L'analyse de graphe permet de détecter des motifs significatifs dans les réseaux de documents, d'identifier les communautés de sujets, de mesurer l'influence d'un document au sein du réseau, et de faciliter la navigation et la découverte d'informations pertinentes dans un grand corpus. Ce travail vise à aider les utilisateurs et utilisatrices de la plateforme *Persée* à exploiter et extraire les informations cachées dans cette bibilothèque scientifique numérique.

1.1 Pistes de recherche

Nous proposons deux pistes de recherche. D'un côté, nous exploitons les informations structurelles et évaluons ce que peut apporter l'analyse de graphe dans l'analyse d'un grand corpus. De l'autre côté, nous utilisons des modèles de langues avancés afin de proposer des solutions performantes pour traiter les données textuelles (Section 5.2 et Section 6.2)

2 Prétraitement et Statistiques descriptives

La première étape du projet consiste à récupérer, nettoyer et sauvegarder les données dans un format adapté aux analyses qu'on se propose d'effectuer dans un second temps. Les données fournies comprenaient des informations textuelles telles que le titre des articles, leur abstract (nous nous sommes limités à la langue française pour ce travail), ainsi que des informations structurelles permettant de relier les documents ou les auteurs entre eux.

Les données sont extraites du portail www.persee.fr, spécialisé dans la diffusion du patrimoine scientifique en sciences humaines et sociales. Le dataset comprend plus de 908 780 documents variés, majoritairement en français. Chaque document est associé à une collection correspondant généralement à une revue scientifique, avec une discipline principale pour la navigation sur le portail. Les données sont structurées en tableaux, avec des colonnes telles que le titre, les auteurs, la date de publication, des abstractS.

Afin de construire un graphe pertinent pour l'analyse, nous gardons uniquement les données qui ont un abstract, ce qui fait au total 99 888 documents. En effet, seuls 11% des documents ont un abstract. Puis nous sélectionnons les documents dont le domaine comprend plus de 4000 documents, ainsi, il nous reste 10 domaines. Nous visulisons la distributions des documents par domaine.

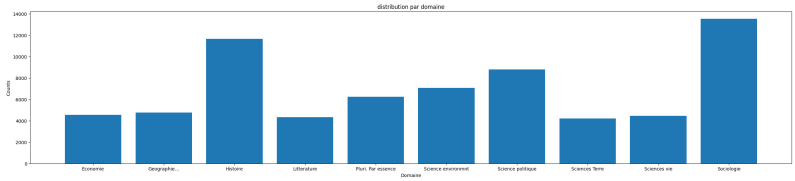


Fig. 1: Distribution des domaines

Puis, pour uniquement le première piste de recherche, 1% de ces documents sont sélectionnés par un tirage aléatoire et stratifié vis-à-vis des domaines. Ce choix de traiter un petit échantillon est pour faciliter l'analyse de graphe sous contraite de capacité computationnelle. Ainsi, nous obtenons un dataset qui se compose de 696 documents qui appartiennent aux 10 domaines. Le titre et le résumé étant concaténés, la taille moyenne des documents est de 985 mots.

Une deuxième étape de prétraitement est nécessaire avant de se lancer dans les analyses textuelles. en effet, nous avons besoin de nettoyer les textes afin d'améliorer la qualité de nos analyses. Il s'agit surtout de réduire la taille du vocabulaire, qui peut occuper une large partie de l'espace mémoire au moment de l'inférence des modèles comme LDA (Latent Dirichlet Allocation, Blei, Ng, and Jordan 2001) et ralentir considérablement les temps de calcul. Pour cela, nous retirons les caractères spéciaux comme les signes de ponctuations et les mots vides de sens (*stopwords*). Il s'agit d'une étape essentielle pour améliorer la qualité des classifications qui viendront juste après.

D'autres prétraitements, cette fois-ci dans le but de faciliter l'interprétation des résultats et la compréhension des données sont effectués. En particulier, le code de la collection à laquelle appartient le document est isolé dans les liens qui servent à

les identifier grâce à des expressions régulières. Le même travail est fait pour isoler l'identifiant (suite de chiffres) des auteurs.

Un nuage de mots-clés à été généré à partir des abstracts nettoyés afin de jeter un rapide coup d'oeil sur le corpus à notre disposition :

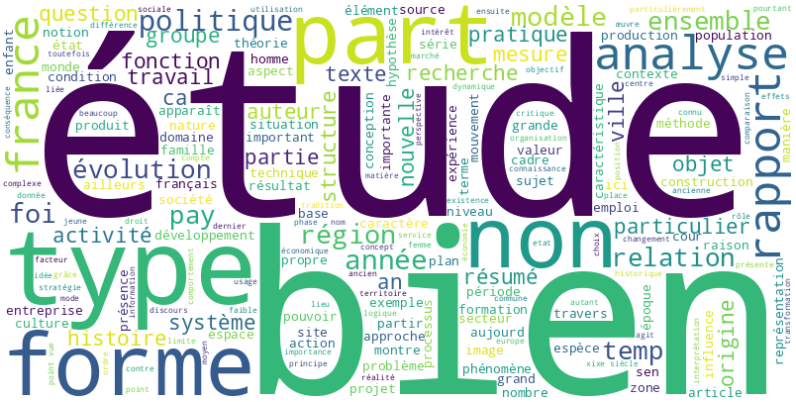


Fig. 2: Nuage de mots-clés des abstracts en français

3 structure du corpus

3.1 Construction du graphe

Nous avons d'abord essayé de construire un graphe sur les 99 888 documents qui ont un résumé à partir des relations naturelles qui existent dans le dataset

- un document A est voisin d'un documents B s'il est cité par B ou il cite B. Densité du graphe : presque 0.
- un document A est voisin d'un documents B si ils citent touts les deux un document C. Densité du graphe : presque 0.
- un document A est voisin d'un documents B s'il ont au moins un auteur commun. Densité du graphe : 8×10^{-4}

Etant donné que notre objectif est d'exploiter l'information structurée et d'évaluer l'apport de cette information par rapport aux analyses standards (sans information structurelle), la densité ultra faible de ces graphe rend l'analyse du graphe non pertinent.

Ainsi, nous avons construit le graphe par une autre approche basée sur les représentations vectorielles donnée par Sentence Bert. En effet, (Reimers and Gurevych 2019) montre que la similarité cosinus de ces représentations vectorielles des documents est comparable avec la similarité sémantique. (Reimers and Gurevych 2019) ont testé cet aspect sur la tâche de Semantic Textual Similarity (STS) et ont obtenu des performances convainquantes. Inspirés de ces résultats, nous calculons donc une matrice d'adjacente non pondérée en utilisant la similarité cosinus. L'arrête entre

deux documents est crée lorsque leur similarité cosinus est supérieur à 0.4. Basé sur la propriété du cosinus, ce seuil est choisi pour avoir une densité de graphe désirée. A part le plus grand composant connexe (536 noeuds sur 696), les 140 petits composants n'ont qu'un ou deux noeuds. Nous observons que les noeuds ne sont pas réparties de

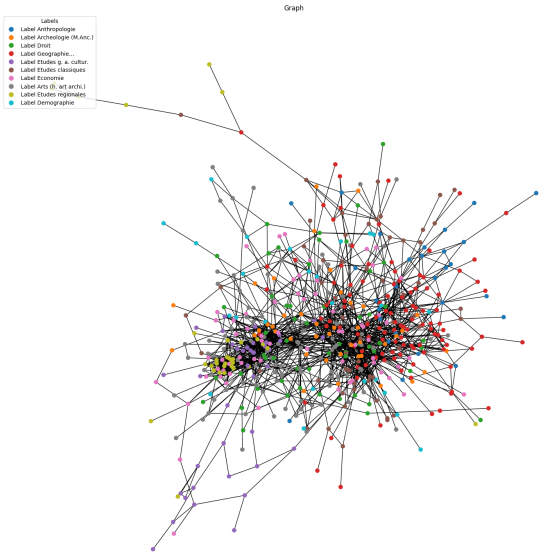


Fig. 3: Le plus grand composant connexe

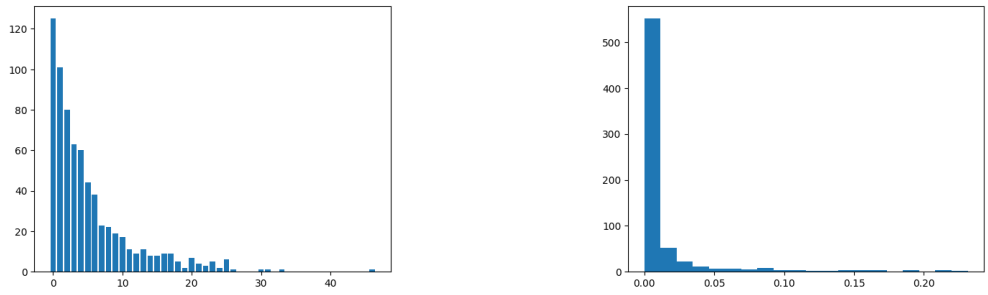
manière aléatoire vis-à-vis des domaines. Cela justifie en partie l'approche adaptée.

3.2 Statistiques du graphe

Le plus grand composant connexe est de densité 7.5×10^{-3} et de diamètre 14. Le degré et la centralité eigenvector (Page Rank centrality) des noeuds sont distributés comme dans la Figure 4. Nous constatons avec ces deux distributions que certains documents ont une mesure de centralité beaucoup plus élevé que les autres. Il s'agit éventuellement des publications qui traitent des sujets généraux ou des revus systématiques d'un domaine.

4 Moteur de recherche

Un moteur de recherche est mise en place, il permet à l'utilisateur de saisir des mots-clés et de récupérer les articles pertinents. Vu que notre graphe est construit à partir des représentations vectorielles données par Sentence Bert. Une solution évidente est de calculer la similarité cosinus de la représentation du texte saisie par l'utilisateur avec tous les noeuds du graphe, puis proposer les documents les plus proches. Cependant, cela peut générer des des documents éloignés pour des raisons différentes. Par exemple,



(a) distribution des degrées

(b) distribution eigenvector

Fig. 4: Mesures de centralité

la saisie n'est pas concise ou elle est intrinsèquement éloignée de tous les noeuds sauf un. Une solution alternative est de prendre en compte à la fois l'information textuelle et les relations structurelles, il s'agit de prendre uniquement le noeud le plus similaire et les voisins de ce noeud. Avec la saisie *Immigration en Europe*, la première solution donne résultats suivants :

- L'Alsace à l'image de l'espace migratoire européen
- Les divergences d'évolution des marchés du travail allemand et européens
- L'accès des jeunes Français à l'emploi.
- La Flandre : un puzzle d'eurorégions et de coopérations transfrontalières
- La nouvelle donne migratoire en Europe du Sud

Et la deuxième solution donne les résultats suivants :

- L'Alsace à l'image de l'espace migratoire européen
- Faire des Français d'Algérie des métropolitains
- Quelle population pour les régions en 2015 ?
- Les migrants de Turquie face à la France, confrontations d'identités
- Les trois "âges" de l'émigration algérienne en France
- L'immigration sur le plateau du Neubourg vers 1804

Curieusement, la deuxième solution donne des résultats beaucoup plus pertinents que la première. Nous ne saurons pas expliquer la raison derrière, et il est possible que les deux solutions convergent lorsque la taille du corpus augmente.

5 Clustering

5.1 Clustering Spectral

Regrouper les documents similaires par le clustering et et comparer les clusters avec les label est une analyse intéressante. Nous avons appliqué le clustering sperctral avec la matrice d'adjacence construite avec la similarité cosinus. Pour illustrer les claustres trouvés, nous les visualisons sur le plus grand composant connexe.

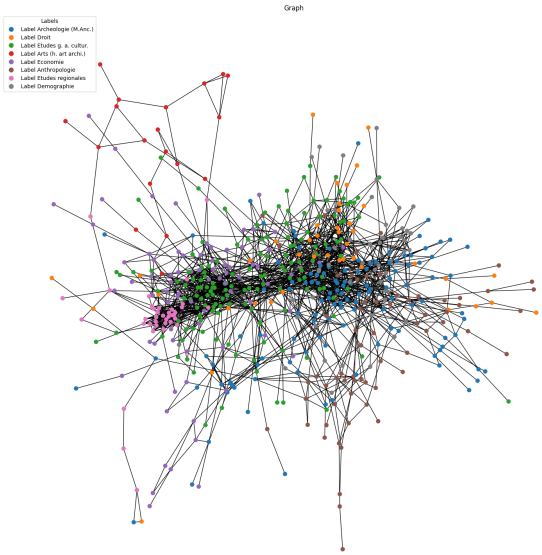


Fig. 5: Clusters par le clustering sperctral

Les noeuds du même domaine se rapprochent plus par rapport à la Figure 3. En effet, cela est dû au fait qu'on utilise aussi la similarité cosinus pour la construction du graphe. Ainsi, la comparaison des deux graphes serait plus intéressante lorsque le graphe est construit différemment.

L'Adjusted Rand Score entre les clusters et les vrais labels est de 0.118, ce qui est assez faible. A ce stade, sous condition que les documents soient labelisés correctement, une question naturelle se pose : est ce que la similarité sémantique entre les publications scientifiques en sciences humaines et sociales dépend peu du domaine ?

5.2 Topic Modeling

Les méthodes de Topic modeling permettent d'explorer et d'analyser de manière non supervisée les données textuelles. En particulier, elles permettent de détecter automatiquement des thématiques, c'est-à-dire des sujets récurrents dans les documents. Ces thématiques peuvent servir plusieurs objectifs : voir ce dont parlent les documents, les annoter, les organiser et les regrouper en clusters. Etant donné que les documents sont déjà regroupés dans des collections, nous allons uniquement nous attacher à explorer les thématiques qui se trouvent dans le corpus. Nous allons pour cela utiliser la méthodes LDA (Latent Dirichlet Allocation, Blei, Ng, and Jordan 2001), un modèle probabiliste génératif utilisé en apprentissage automatique pour la découverte de topics cachés dans un ensemble de documents. L'idée fondamentale derrière LDA est que chaque document peut être décrit par une distribution sur un ensemble fixe de topics, et chaque topic peut être caractérisé par une distribution sur l'ensemble des mots du vocabulaire. Nous avons entraîner sur l'emsemble des abstracts en français un modèle

LDA avec un nombre fixe ($k = 25$, afin d'avoir à la fois des thématiques précises et pour pouvoir les visualiser facilement) topics. Grâce à la méthode *LDAVis* (Sievert and Shirley 2014), qui génère un visualisation interactive des modèles LDA. Elle permet de mieux comprendre et interpréter les résultats de ces modèles en proposant une interface graphique intuitive. LDAVis affiche les topics trouvés par LDA sous forme de bulles dans un espace bidimensionnel. La proximité entre les bulles reflète la similarité entre les topics, aidant à comprendre comment les différents topics sont liés entre eux. De plus, pour chaque topic, LDAVis montre une liste des mots les plus pertinents, ainsi que leur contribution à ce topic. Cela permet d'identifier rapidement les caractéristiques principales de chaque topic. Enfin, l'interface permet de sélectionner un topic pour examiner en détail sa composition et de voir comment les mots contribuent à ce topic. On peut également ajuster le paramètre de pertinence pour filtrer les mots en fonction de leur spécificité au topic. Voici un bref aperçu du rendu :

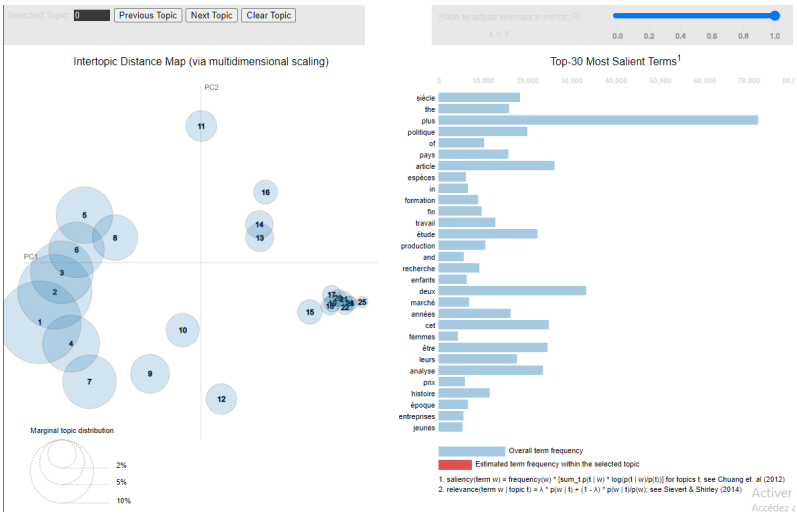


Fig. 6: Aperçu de l'interface générée par LDAVis à partir des 25 thématiques trouvées dans les abstracts en français

Les thématiques trouvées par la méthode LDA sont facilement exploitables car elles sont très parlantes et assez différentes les unes des autres. Par exemples, la première thématique évoque l'utilisation de méthodes scientifiques, avec par exemple les mots *analyse*, *données*, *résultats*, *processus* et *étude*. Si on observe à présent la thématique 4, on retrouve des mots qui suggèrent une thématique à la démographie et à l'urbanisation avec des mots comme *espace*, *développement*, *ville*, *territoire*, *urbain*, etc... Pour finir, la thématique 14 est sans équivoque, avec des mots très en lien avec la religion catholique : *église*, *roi*, *saint*, *Dieu*, *évêque* ou encore *Vatican*. Sans explorer chacune des 25 thématiques, on remarque que la méthode LDA est particulièrement utile pour comprendre la structure de notre corpus, et pour avoir une première idée des principaux sujets qui y sont abordés.

6 Classification

6.1 l'information textuelle et les relations structurelles

La tâche consiste à prédire la discipline associée aux documents. Ici, nous aimerions évaluer l'apport de l'information structurelle pour cette tâche et ainsi de valoriser l'approche de considérer les documents comme les noeuds d'un graphe.

Pour ce faire, nous utilisons les représentations vectorielles du Sentence Bert. Deux modèles ont été testé : l'un est un réseau de neuronnes à deux couches linéaire et l'autre est un Graph Convolutional Network à deux couches accompagnées d'une couche linéaire. Et nous faisons en sorte que l'entraînement d'un modèle n'est pas plus abouti que l'autre. Les premier résultats obtenus sont assez mauvais, l'accuracy sur le jeu de test (20 % et stratifié vis-à-vis du label) pour le modèle simple est de 0.486 et pour le GCN 0.429.

D'abord, il faut signaler que la faible performance peut être à cause du nombre insuffisant des données, du sur-apprentissage, ou d'autres problèmes que nous ne voulons pas traiter spécialement dans le cadre de ce projet. En revanche, le fait que l'ajout de l'information structurelle a baissé l'accuracy remet en question la manière dont nous construisons le graphe. Nous essayons d'améliorer la performance en nous inspirant des statistiques du graphe de la section 2. Nous suggérons que les noeuds qui ont un degré très élevé rend le *text messaging* du GCN moins efficae, parce que par ces noeuds-là, les noeuds qui n'ont en réalité pas beaucoup de similarité vont être connectés et cela peut nuire à l'agrégation des features des voisins. Ainsi, nous avons ré-entrainé un GCN avec un nouveau graphe où les arrêtes des noeuds qui ont un degré supérieur à 10 sont enlevées. Mais finalement, l'accuracy au test n'a pas changé. Cela suggère que notre graphe n'est peut-être pas pertinent vis-à-vis de cette tâche de classification du domaine. Nous reposons ainsi la question qu'on s'est posée dans la section précédente : est ce que la similarité sémantique entre les publications scientifiques en sciences humaines et sociales dépend peu du domaine ? C'est à dire que les deux documents venant de deux domaines différents peuvent avoir une similarité sémantique élevée.

6.2 Classification sur un corpus conséquent

Nous allons à présent tenter d'évaluer l'apport de l'information textuelle seule pour prédire la collection à laquelle appartient un document. Il existe en NLP de nombreuses méthodes pour faire de la classification supervisée : de celles qui reposent sur la construction d'une matrice termes-documents aux méthodes plus récentes basées sur l'architecture Transformers (Vaswani et al. 2023) en passent par l'approche skip-gram, chacune présente des qualités et des inconvénients qu'il faut savoir maîtriser avant de les implémenter. De notre côté, étant donné notre faible puissance de calcul, nous nous sommes penchés sur le modèle FastText (Bojanowski et al. 2016), développé par Facebook Research, et conçu pour la classification de texte et la génération de vecteurs de mots. Il se distingue de la méthode Word2Vec (Mikolov et al. 2013) par sa capacité à prendre en compte non seulement les mots entiers mais aussi les sous-chaînes de caractères, ou "char n-grammes", des mots. Cette approche permet à FastText de

mieux gérer les mots rares, les fautes d'orthographe et les mots nouveaux qui ne figuraient pas dans les données d'entraînement. Pour classer un nouveau document ou phrase, FastText somme les vecteurs de toutes les caractéristiques (mots et n-grammes de caractères) présentes dans le texte. Cette somme de vecteurs est ensuite passée à travers une ou plusieurs couches linéaires suivies d'une fonction softmax pour prédire la probabilité que le texte appartienne à chaque catégorie possible. L'algorithme de FastText étant écris en C, cela en fait un modèle particulièrement rapide à entraîner. C'est pourquoi nous l'avons retenu pour prédire la collection à laquelle appartiennent les documents. Après avoir entraîné le modèle sur une concaténation du titre de l'article scientifique avec l'abstract, afin de conserver un maximum d'informations, nous obtenons un score de précision de 60%, ce qui est meilleur que lorsqu'on ajoute l'information de graphe, mais reste très insuffisant. Notons toutefois que la tâche consistait pour le modèle à prédire une parmi les 27 classes possibles. Il s'agit donc d'une tâche bien plus complexe que s'il s'agissait d'une classification binaire. De plus, rappelons que 89% des documents n'ont pas d'abstract, donc le modèle n'avait que le titre, dont certains stop words ont été retirés, pour s'entraîner et faire une prédiction. Le score de 60% n'est donc pas très surprenant et même plutôt encourageant pour poursuivre ce travail vers des méthodes plus sophistiqués, comme par exemple des LLM type Bert.

7 Conclusion

Pour conclure, ce travail sur la base de données Persée nous a permis de mettre en oeuvres plusieurs techniques en analyses de graphes et en NLP. En ce qui concerne l'analyse de graphe dans l'analyse du corpus structuré, même si nous pouvons extraire des informations sur le corpus telles que les documents centraux et les clusters des documents, nous n'avons pas pu mieux réussir la tâche de classification à l'aide des relations structurelles. Cependant, elles nous permettent de créer un moteur de recherche pertinent. Il faudra mieux exploiter cette information riche qui permettra sûrement de mieux utiliser les grands corpus. Concernant la deuxième piste de recherche qui consiste à utiliser les modèles de langues avancés, après avoir effectué quelques pre-traitements sur le texte, notamment pour alléger le vocabulaire en supprimant les mots-outils, nous nous sommes penchés sur deux approches : une non supervisée avec LDA afin de détecter les principales thématiques présentes dans le corpus. Cette méthode s'est révélée particulièrement efficace pour avoir un aperçu de qualité sur les différents sujets évoqués dans les articles scientifiques. La deuxième approche, supervisée, consistait à prédire la discipline à laquelle appartenait un document. L'utilisation des GCN s'est révélée infructueuse, mais FastText, qui apprend des représentations en grande dimension des mots et des n-gram de caractères, s'est montré un peu meilleur dans cette tâche, bien que l'information textuelle soit limitée par le faible nombre d'abstracts dans le corpus.

References

Blei, David, Andrew Ng, and Michael Jordan (Jan. 2001). "Latent Dirichlet Allocation". In: vol. 3, pp. 601–608.

Bojanowski, Piotr et al. (2016). "Enriching Word Vectors with Subword Information". In: *arXiv preprint arXiv:1607.04606*.

Mikolov, Tomas et al. (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv: 1301.3781 [cs.CL].

Reimers, Nils and Iryna Gurevych (2019). "Sentence-bert: Sentence embeddings using siamese bert-networks". In: *arXiv preprint arXiv:1908.10084*.

Sievert, Carson and Kenneth Shirley (June 2014). "LDAvis: A method for visualizing and interpreting topics". In: DOI: 10.13140/2.1.1394.3043.

Vaswani, Ashish et al. (2023). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].