Indexation de flux télévisuel grâce aux microblogs

Charles Robin

Master de Recherche en Informatique

Rapport de Stage

Indexation de flux télévisuel grâce aux micro-blogs

Auteur :
Charles Robin

Superviseur :
Vincent Claveau
TEXMEX

Résumé

L'indexation vidéo télévisuelle se fait habituellement par extraction de caractéristiques depuis le flux vidéo mais ces techniques ne permettent pas d'obtenir un haut niveau sémantique. Nous proposons d'extraire des informations sémantiquement intéressantes directement depuis des microblogs commentant ce flux vidéo ainsi que du texte journalistique.

Dans cette optique nous voyons les techniques existantes d'extraction d'informations. Nous nous intéressons à la reconnaissance d'entités nommées, à la détection d'évènements et à l'analyse de sentiments. Puis nous nous intéressons aux différentes particularités des microblogs par rapport à d'autres sources textuelles comme le texte journalistique, comme leur taille ou leur niveau de langage. Nous proposons une méthode pour utiliser les avantages des deux sources de données, en combinant la qualité du texte journalistique ainsi que des informations de popularité tirées des microblogs, en utilisant des techniques éprouvées d'alignement de séquences et de recherche d'information. Puis, nous nous intéressons ensuite à l'analyse de la polarité dans ces microblogs. Nous obtenons des assez bons résultats qui ouvrent beaucoup de pistes pour de travaux futurs.

Remerciements

Dans un premier temps, je tiens à remercier sincèrement M. Vincent Claveau pour m'avoir encadré, mais également pris part activement à la réussite de ce stage. Je tiens aussi à le remercier pour toute l'aide qu'il a pu m'apporter à la fois dans l'écriture de la bibliographie et tout au long de mon stage.

Je tiens aussi à remercier toute l'équipe TEXMEX de l'IRISA et en particulier les autres stagiaires de mon bureau à savoir Ahmet, Caio, Miaojing, Thomas et Vedran qui ont contribué au succès de ce stage et qui m'ont permis de m'intégrer facilement.

Table des matières

Chapitre 1

Introduction

Depuis plusieurs années, on assiste à une multiplication du nombre de vidéos, qu'elles soient tournées en amateur pour une utilisation personnelle ou qu'elles soient créées dans un but de diffusion par des professionnels. Par exemple, plus de 100 heures de vidéo sont ajoutées chaque minute sur la plate-forme Youtube [1]. L'indexation de vidéos par le contenu devient nécessaire pour parcourir, rechercher, résumer, et manipuler ces vidéos. Il est donc nécessaire de générer de façon automatique les descriptions les plus informatives possible de ces vidéos.

Dans ce but, beaucoup de travaux ont cherché à générer des descripteurs à partir du signal vidéo directement. Ces caractéristiques peuvent être visuelles, par exemple des histogrammes de couleurs, des vecteurs mouvements, ou des descripteurs de formes, ou bien elles peuvent se baser uniquement sur la modalité sonore de la vidéo en utilisant par exemple le niveau d'énergie, le Zero-Crossing Rate du signal sonore ou les plages de silence [Baghdadi, 2010].

Ces solutions n'offrent cependant pas une description sémantiquement intéressante pour beaucoup d'applications impliquant une interaction avec un utilisateur. Il existe une différence sémantique entre le contenu de la vidéo tel que perçu par l'utilisateur et les descripteurs générés. On parle alors de *fossé sémantique*. Nous cherchons donc à remédier à ce problème en utilisant comme source d'information des indices de plus haut niveau sémantique.

Dans le cadre de ce stage, nous nous intéressons à un type de vidéos particulier, les flux télévisuels. Ces vidéos sont souvent commentées par les utilisateurs des réseaux sociaux, et en particulier des utilisateurs de Twitter [2], réseau social de *microblogging*. Nous allons utiliser ces microblogs comme source d'informations. Ces textes sont d'assez haut niveau sémantique car ils sont très majoritairement rédigés par des humains à destination d'autres personnes. Ils contiennent les informations jugées pertinentes par l'auteur, résumées d'une manière concise.

1. `http://www.youtube.com/yt/press/fr/statistics.html`
2. http ://twitter.com

Dans le cadre ce stage, nous utilisons comme contexte applicatif les matchs de football, que nous indexons grâce aux microblogs du réseau social Twitter, appelés *tweets* . Nous travaillons uniquement sur des *tweets* en langue française. Le sport, et en particulier les matchs de football, sont particulièrement représentés sur Twitter : 7 des 10 événements ayant généré le plus de trafic sur le réseau social en France en 2013 sont des événements sportifs [3]. Par exemple, le match de football *France-Ukraine* du 19 novembre 2013 a été relaté dans plus de 1,2 million de *tweets*. En plus du nombre de *tweets* générés à chaque match, ce contexte applicatif est particulièrement intéressant pour les raisons suivantes :

- de nombreux travaux ont déjà étudié les matchs de football en utilisant des descripteurs de signal [Yow et al., 1995, Barnard et al., 2003] ;
- de par la popularité du sport, de nombreux débouchés sont possibles. On peut penser à plusieurs applications, comme une collection interactive de matchs de football ;
- les matchs de football sont très peu structuré (uniquement deux mi-temps), l'indexation y est beaucoup plus difficile que d'autres évènements sportifs (matchs de tennis).

Dans la chapitre 2, nous abordons plus en détail le but du stage ainsi que le résultat que nous voulons atteindre. Puis, dans le chapitre 3, nous présentons un état de l'art des travaux existants, en particulier sur l'extraction d'informations et les opérations sur les microblogs. Nous décrivons dans le chapitre 4 la constitution de notre corpus de données. Nous exposons dans le chapitre 5 la synchronisation des différentes sources d'informations que nous avons réalisée. Le chapitre 6 est consacré à l'étude de la polarité dans les *tweets* . Nous nous intéressons dans le chapitre 7 aux possibilités de travaux futurs. Enfin, nous concluons ce document dans la section 8.

3. `https://blog.twitter.com/fr/2013/lannee-2013-sur-twitter`

Chapitre 2

Présentation du sujet

Le but de ce stage est d'indexer un flux télévisuel à partir de microblogs. Nous devons extraire des informations de cette source d'informations et la synchroniser avec le flux télévisuel. Nous nous fixons comme cadre applicatif d'indexation de générer plusieurs nuages de mots ordonnancés chronologiquement tout au long du match. Un exemple d'un résultat obtenu peut être visible à la figure 2.1. Chaque moment du match est associé à une description permettant à l'utilisateur de naviguer facilement, ou à un système de proposer d'autres traitements (par exemple un résumé automatique). Dans notre cadre applicatif du football, nous nous basons sur deux types de sources d'informations complémentaires, les *tweets* et les résumés minutes-par-minutes, présentés respectivement à la section 2.1 et 2.2. Nous souhaitons pouvoir utiliser les avantages de chacune de ces deux sources d'informations, afin d'utiliser ces informations pour l'indexation du flux vidéo.

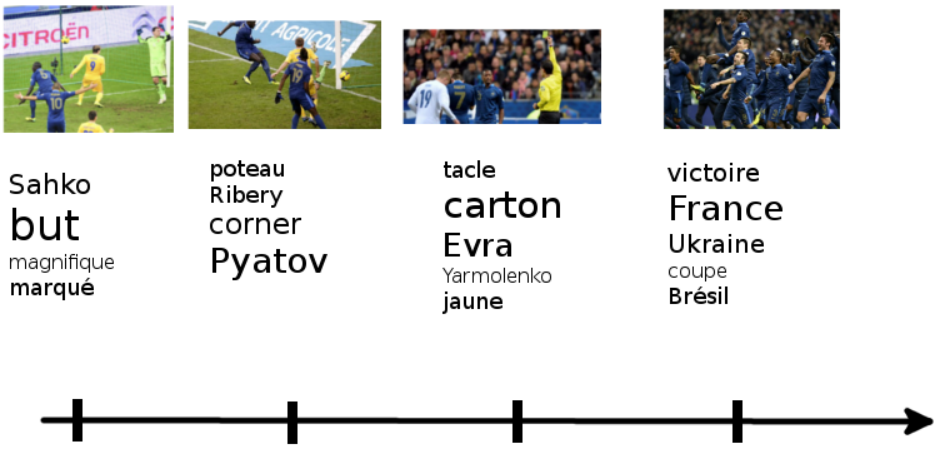

FIGURE 2.1 – Exemple d'un résultat attendu

FIGURE 2.2 – Exemple d'un *tweet* concernant le match France-Ukraine.

2.1 Présentation de Twitter et des *tweets*

Les *tweets* sont les microblogs du réseau social Twitter. dans lequel ses utilisateurs peuvent lire et écrire des messages très courts (140 caractères maximum). Ce réseau social compte à l'heure actuelle plus de 255 millions de membres actifs [1], et plus de 4.5 millions de visiteurs uniques en France [2]. Un exemple de *tweet* est visible à la figure 2.2.

Les *tweets* peuvent contenir des mots-dièses (*hashtag*) qui sont des mots commencent par un croisillon (#). Les mots-dièses sont des mots-clés décidés par l'auteur, permettant de relier le *tweet* à d'autres *tweets* portant sur la même thématique. Sur le *tweet* en exemple à la figure 2.2, le mot *#FRAUKR* est un mot-dièse (pour FRAnce - UKRaine), qui permet à ce *tweet* d'être relié à toutes les autres publications comportant ce mot-dièse.

2.2 Présentation des résumés minutes par minutes

Les résumés minutes-par-minutes sont écrits par des journalistes qui indiquent les actions qu'ils jugent les plus marquantes des matchs. Ils sont produits en direct lors des matchs de football. Étant écrits par des journalistes, les résumés minutes-par-minutes sont donc de bien meilleure qualité que les *tweets* en ce qui concerne la qualité de la langue (orthographe, syntaxe, typographie, vocabulaire). En revanche, les résumés ne donnent aucune information d'importance entre les différentes minutes qui les composent. Dans la suite de ce rapport une minute sera une ligne d'un résumé minutes-par-minutes.

Un exemple de résumé minutes-par-minutes est visible à la table 2.1.

1. https ://investor.twitterinc.com/releasedetail.cfm ?ReleaseID=843245
2. http ://www.lefigaro.fr/secteur/high-tech/2013/06/24/01007-20130624ARTFIG00270-les-francais-se-sont-massivement-convertis-aux-reseaux-sociaux.php

Minute	Action
...	
67	Carton jaune pour Evra pour un tacle par derrière sur Yarmolenko.
69	Carton jaune pour Debuchy, lui aussi pour un tacle à retardement sur Yarmolenko.
70	Les fautes se multiplient dans le camp français. C'est au tour de Cabaye d'être sanctionné.
71	Libre de tout marquage à trente mètres plein axe, Cabaye tente sa chance. Pyatov est contraint de détourner en corner...
72	But de Sakho ! Suite au corner obtenu par Cabaye, Evra tente une demi-volée du gauche, détournée. Le deuxième centre est le bon. Malgré le marquage de Gusev, Sakho catapulte le ballon du droit dans le but de Pyatov.
...	

TABLE 2.1 – Extrait du résumé minutes par minutes du match France-Ukraine

Chapitre 3

État de l'art

Dans ce chapitre, nous faisons un court état de l'art concernant, dans un premier temps, quelques tâches d'extraction d'informations et dans un second temps, les particularités des microblogs en tant que sources d'informations.

3.1 Extraction d'informations

Dans cette section, nous revenons dans un premier temps sur les pré-traitements utilisés en extraction d'informations, puis nous présentons les différentes tâches d'extraction d'informations étudiées.

3.1.1 Pré-traitements usuels

Un pré-traitement usuel utilisé en traitement du langage naturel est l'étiquetage du texte étudié en partie du discours (POS-tagging pour *Part of Speech* tagging), appelé parfois étiquetage morpho-syntaxique. Il consiste à étiqueter chacun des mots du texte étudié par sa partie du discours. Le POS-tagging est utilisé comme pré-traitement dans de nombreuses applications : reconnaissance vocale, génération vocale etc.

L'étiquetage peut permettre de réduire le problème de la polysémie en différenciant des homonymes comme le verbe *dîner* (manger) et le nom *dîner* (repas). Un exemple de POS-tagging en français est présenté ci-dessous :

```
N   V   V    P   D   N   P   N   P   N   P   N   PUNC

Sahko  a marqué dans le but de Pyatov au Stade de France  .
```

La partie du discours ainsi générée est utilisée en tant qu'attribut par les différents outils d'apprentissage artificiel afin de prendre une décision.

De nombreux outils existent afin d'étiqueter du texte, en anglais comme en français. On peut citer le *Stanford Log-linear Part-Of-Speech Tagger*[1] ou *Apache OpenNLP*[2]. Le domaine semble assez mature, des travaux ont réalisé avec succès de l'étiquetage sur des données diverses, allant du texte journalistique aux textes très bruités [Gimpel et al., 2010].

Un autre pré-traitement usuel utilisé pour traiter le langage naturel est la lemmatisation. Il s'agit d'étiqueter chaque terme par son lemme, c'est-à-dire une forme canonique du mot. Par exemple, on utilisera comme forme canonique pour un verbe conjugué le même verbe à l'infinitif, le masculin singulier pour un adjectif, etc. Le lemme obtenu est aussi utilisé en tant qu'attribut pour les classifieurs. La lemmatisation permet d'uniformiser le texte, en remplaçant les diverses formes d'un même mot par un seul et unique mot. Une phrase composée uniquement des lemmes est présentée ci-dessous :

```
Sahko marquer dans le but de Pyatov à+le stade de France.
```

3.1.2 Reconnaissance d'entités nommées

Les entités nommées sont des types d'unités lexicales représentant des entités du monde concret ayant un nom (généralement un nom propre ou un acronyme). Dans le cadre du football, des exemples d'entitées nommées sont les noms des joueurs (Zidane, Platini ...), des noms d'équipes (Manchester United, PSG ...), des noms de stades (Stade de France, Parc des princes ...).

La reconnaissance d'entités nommées (NER pour *Named Entity Recognition*) est une tâche de l'extraction d'informations qui existe depuis plusieurs années. Depuis les années 1990, la NER connaît un fort développement suite à au développement de la recherche d'informations textuelles. La NER consiste à classer des objets textuels, comme des mots ou des groupes de mots, à différentes classes comme par exemple des noms de personnes, d'organisations, ou de lieux.

Il existe trois types de systèmes permettant la NER :
- Les systèmes experts, basés sur des règles [Krupka and Hausman, 1998]
- Les systèmes basés sur l'apprentissage automatique [McCallum and Li, 2003, Liu et al., 2011, Ritter et al., 2011]
- Les systèmes hybrides

Les systèmes experts obtiennent de bons résultats, mais sont réglés pour un jeu de données particulier et sont rarement adaptables. De plus, ils nécessitent la création manuelle de règles, et sont donc coûteux en temps. Les systèmes basés sur l'apprentissage automatique nécessitent une grande quantité de données déjà annotées, afin de reconnaître les formes possibles d'entités nommées.

1. http ://nlp.stanford.edu/software/tagger.shtml
2. http ://opennlp.apache.org/cgi-bin/download.cgi

Récemment, les systèmes basés sur l'apprentissage automatique à base de CRF (pour *Conditional Random Fields*) ont donné de très bons résultats. Les CRF sont des modèles statistiques conditionnels permettant l'étiquetage de séquences de mots. Pour chaque mot, les CRFs utilisent les étiquettes attributs, qui peuvent être le lemme, la partie du discours etc.., du mot courant ainsi que du contexte immédiat [Lafferty et al., 2001]. Les ensembles d'étiquettes B,I,O (pour *begin*, *inside*, *outside*) sont utilisés afin de pouvoir capturer les entités nommées sur plusieurs mots.

Un exemple de la tâche pourrait être le texte suivant, après l'application d'un système de reconnaissance d'entités nommées :

```
B_Personne O    O      O   O   O   O B_Personne  O B_Lieu I_Lieu I_Lieu O

  Sahko    a marqué dans le but de   Pyatov   au  Stade    de   France .
```

La plus grande des limites de ces systèmes basés sur les CRF est la dépendance à la séquentialité du texte. L'étiquette étant calculée à partir des différents attributs des mots aux alentours du mot courant, il est évident que ces systèmes dépendent des variations syntaxiques ou paradigmatiques.

Il existe de nombreux outils disponibles permettant la NER, et ce pour plusieurs langues. La plupart de ces outils sont basés sur les CRF vus précédemment. On peut citer par exemple le *Stanford Named Entity Recognizer*[3] disponibles en plusieurs langues.

Cette technique ayant donné de bons résultats pour une grande variété de tâches, elle semble appropriée pour notre problème. De plus, cette technique est utilisable pour reconnaître les circonstants, les actions, et les autres éléments de ce type à extraire, bien qu'étant originellement développée uniquement pour la NER. En effet, elle requiert uniquement un corpus annoté, ainsi que certains pré-traitements. Nous revenons en section 3.2 sur certaines limites qui peuvent apparaître à cause de la spécificité des textes à traiter.

3.1.3 Détection des événements

Bien que la notion d'événement ne suive pas une définition absolue, il est admis qu'un événement est causé dès lors qu'il y a modification d'un état. Un événement est ancré dans le temps, et est associé à une durée plus ou moins longue [Arnulphy, 2012]. Dans notre cadre applicatif du football, les événements peuvent être la sanction d'un joueur par un carton (jaune, rouge), l'évolution du score après un but d'une équipe etc.

La détection des événements est une sous-tâche de l'extraction d'informations. Elle permet de savoir à quel moment se passe un événement d'intérêt du match. Dans notre cas applicatif, la détection d'événements peut servir à synchroniser temporellement les différents acteurs trouvés lors de la section précédente au flux vidéo.

3. http ://nlp.stanford.edu/software/CRF-NER.shtml

[Lanagan and Smeaton, 2011] proposent dans leur article de détecter les événements d'intérêt en utilisant le flux de *tweets*. Leur solution est rapide et n'utilise pas le flux vidéo.

Les auteurs génèrent, grâce au débit de *tweets*, des fenêtres de vidéos. Ils sélectionnent ensuite les fenêtres dont le nombre de *tweets* dépasse un seuil donné. Les fenêtres ainsi trouvées contiennent alors un événement.

Cet article est intéressant pour notre étude, mais les auteurs ne s'intéressent qu'aux plans contenants ou non un événement. De plus, les auteurs n'utilisent pas du tout le contenu, mais uniquement du débit des *tweets*. Il pourrait être intéressant de coupler ces travaux à du traitement automatique du contenu de ces microblogs. Les auteurs ne s'intéressent qu'à l'apparition ou non d'un événement ; pour notre part, nous souhaitons obtenir un peu plus de sémantique : de quel événement s'agit-il, qui est à l'origine de cet événement etc... Toutefois, nous allons pouvoir nous inspirer de cet article afin de détecter les événements et à les synchroniser avec les différents résultats discutés à la section précédente.

3.1.4 Détection d'opinion

La détection d'opinion, ou l'analyse de sentiments, permet d'analyser le point de vue de l'auteur d'un texte. De la même façon que pour la reconnaissance d'entités nommées, beaucoup d'approches utilisent des dictionnaires comme des adjectifs péjoratifs ou laudatifs. Cependant, certaines techniques sont basées sur de l'apprentissage artificiel [Pang et al., 2002, Thelwall et al., 2010]. L'analyse de sentiments est une tâche assez difficile de par la nature de ce que l'on veut reconnaître. En effet, le point de vue de l'auteur peut varier selon les personnes, et ce beaucoup plus que la tâche de reconnaissance des entités nommées.

Dans l'objectif de l'indexation vidéo, l'analyse de sentiments peut nous servir à mesurer le degré de subjectivité des textes, et donc nous permettre de mesurer la pertinence du texte et des informations qu'il contient. Cela peut nous donner une idée de la confiance à accorder à l'information extraite de ce texte. Il peut aussi être intéressant de générer des résumés-minutes subjectifs et de mesurer la subjectivité de ceux-ci.

Une approche simpliste de la tâche peut être vue comme une catégorisation des textes en plusieurs classes, représentant les différentes opinions possibles (laudative, péjorative, objective). [Thelwall et al., 2010] proposent un système capable d'évaluer à quel point un texte est laudatif ou péjoratif, en se basant sur une échelle de 1 à 5.

De la même façon que pour les NER, il existe des systèmes à base de dictionnaires et des systèmes basés sur l'apprentissage automatique afin de détecter les sentiments et classer les textes [Sebastiani, 2002]. On peut citer un des dictionnaires les plus utilisés en détection d'opinion *SentiWordNet*[4]. Cependant, les travaux récents ont de meilleurs résultats avec des approches hybrides ou

4. `http://sentiwordnet.isti.cnr.it/`

à base d'apprentissage automatique. En effet, les dictionnaires étant limités, de nombreux exemples ne contiennent aucun des mots du dictionnaire.

[Pang et al., 2002] ont comparé plusieurs classifeurs pour la détection d'opinion : un classifieur naïf bayésien, un modèle à maximum d'entropie et un SVM [5]. Ils ont aussi utilisé ces classifieurs sur différents ensembles d'attributs comme des n-grammes ou la partie du discours. Les résultats sont très proches pour tous les classifieurs utilisés.

3.2 Cas particulier des microblogs

Dans cette section nous présenterons les particularités des microblogs en tant que source d'informations textuelle. Nous présentons en premier lieu les difficultés liées à la taille des messages, puis celles reliées au niveau de langue.

3.2.1 Problèmes relatifs à la taille du texte

Les microblogs sont des textes écrits par les utilisateurs de réseaux de microblogging qui ont pour principale caractéristique d'être très courts. Par exemple, les *tweets* ont une longueur inférieure à 140 caractères. Cette caractéristique rend les microblogs difficiles à analyser. En effet, nous ne disposons que de peu de contexte vis-à-vis des *tweets*. La taille réduite des *tweets* cause de nombreuses ambiguïtés, ce qui peut nuire à la compréhension d'une personne qui ne possède pas le contexte du microblog, mais plus encore à des systèmes d'extraction d'informations.

La contextualisation des microblogs est donc une étape importante si l'on veut récupérer le contexte. La contextualisation de microblogs est un nouveau domaine et s'apparente à la génération d'un texte résumant le contexte. Quelques travaux récents abordent ce problème, par exemple l'article de [Deveaud et al., 2013] réalisé dans le cadre de la tâche *Tweet Contextualisation* d'INEX [6]. Les auteurs utilisent des articles de l'encyclopédie en ligne Wikipédia afin de récupérer du contexte autour de ces *tweets*. Ils sélectionnent tout d'abord les meilleurs articles à partir des différents termes du *tweet*. Puis ils sélectionnent un certain nombre de phrases de ces articles en fonction de la pertinence de ces phrases par rapport aux articles et aux termes du *tweet*.

3.2.2 Problèmes relatifs à qualité de langue

Les microblogs ne sont pas seulement courts, ils sont aussi bruités. Le style d'écriture de certains utilisateurs peut parfois être incompréhensible pour les non-initiés. Les utilisateurs commettent souvent des fautes de grammaire, d'orthographe ou de frappe. De plus, ils utilisent des abréviations, des mots propres aux réseaux sociaux, des onomatopées, des symboles, des néographies (*koi* = quoi).

5. Séparateurs à Vaste Marge, ou *Support Vector Machine*
6. `https://inex.mmci.uni-saarland.de/tracks/qa/`

Dans le cadre de leur article, [Ritter et al., 2011] ont réalisé une petite expérience en utilisant les outils de l'état de l'art (*Stanford Named Entity Recognizer*) afin de reconnaître les NER sur des *tweets* . Un exemple est visible ci-dessous :

```
   B_Organisation O B_Organisation O  O      O     B_Organisation     O       O
        Yess      !      Yess      ! Its official     Nitendo     announced today
  O
that

    O    O      O     O  B_Organisation  O   O    O    B_Lieu    O    O  O    O
   they Will release the     Nitendo     3DS in north America march 27 for $250
```

On remarque que le mot $Yess$, probablement en dehors du vocabulaire des mots de l'ensemble d'apprentissage a été reconnu comme une organisation, alors qu'il s'agit simplement d'un dédoublement de la consonne finale. De plus, bien que la première occurrence du mot $Nintendo$ ait été bien étiquetée, la deuxième occurrence n'a pas été segmentée correctement. Le résultat attendu était le produit $Nintendo3DS$. Enfin, l'omission de la majuscule sur le mot *north* a causé la non-reconnaissance de l'entité $northAmerica$. On remarque que les majuscules semblent avoir un grand impact sur la NER, or elles sont souvent omises dans les tweets.

Les outils existants sont souvent entraînés généralement sur du texte journalistique, et ont donc de mauvais résultats sur les *tweets*. Il en est de même pour de nombreuses autres tâches du traitement du langage, comme l'étiquetage en partie du discours ou la détection d'opinions.

La différence de niveau de langue implique de nombreuses variations qui vont nuire aux tâches d'apprentissage. Les variations paradigmatique ou anaphorique [Daille, 2002] sont les plus complexes à traiter. Elles posent le problème du paraphrasage ou de la synonymie. Dans notre contexte, ce problème peut être illustré par les exemples suivants désignant tous l'action de marquer un but :

- ouvrir le score
- marquer un but
- remporter son duel avec le gardien

Chapitre 4

Constitution du corpus de données et environnement de développement

Dans ce chapitre, nous décrivons la récupération du corpus de données que nous avons effectuée. Nous décrivons dans un premier temps ce corpus, et dans un deuxième temps, nous exposons l'environnement technique de récupération de ces données.

4.1 Description du corpus

Dans un premier temps, nous avons dû constituer notre corpus de données. Nous avons pour cela récupéré directement à partir du site de Twitter des ensembles de *tweets* décrivant plusieurs matchs. Nous avons dû aussi récupérer des résumés minutes par minutes à partir de 3 sites différents : *Football365*[1], *Le monde Football*[2] et *L'Équipe*[3]. Nous avons récupéré les ensembles de *tweets* et de minutes pour 13 matchs de football professionnel, impliquant dans tous les cas deux équipes françaises.

4.1.1 Descriptions des *tweets*

Nous récupérons les *tweets* contenant certains types de *hashtag*. Pour un match donné entre deux équipes *foo* et *bar*, nous capturons tous les *tweets* contenants l'un des *hashtags* suivants *#foo*, *#bar*, *#foobar*, *foo* et *bar* pouvant soit représenter le nom de la ville (Paris), soit le nom de l'équipe (PSG). Nous considérons que pour chaque match les noms des équipes et des villes sont connus, en effet, ils peuvent être récupérés directement à partir de sites spécialisés, ou de bases de données publiques comme *football.db*[4]. Un tableau récapitulatif du nombre de ces *tweets* peut être vu à la table 4.1.

1. http ://www.football365.fr/
2. http ://www.lemonde.fr/football/
3. http ://www.lequipe.fr/Football/
4. http ://openfootball.github.io/

Maximum	Minimum	Moyenne	Écart-type
31 699	374	9 174	10 187

TABLE 4.1 – Nombre de *tweets* par match

Nous récupérons pour chaque *tweet*, en plus du contenu du message, la date et l'heure à laquelle le *tweet* a été posté, le nom de l'auteur ainsi que l'identifiant du *tweet*. Le nom de l'auteur nous permet dans quelques cas de déterminer la polarité du *tweet* (voir section 6). Nous récupérons la date et l'heure du *tweet* afin de pouvoir synchroniser temporellement ces *tweets* au reste des données.

Une des premières remarques que nous pouvons faire est la grande différence de *tweets* en fonction des matchs. En effet, certains matchs sont plus populaires que d'autres. Par exemple, le match *Paris Saint Germain - Olympique de Marseille* a suscité plus de 31 500 *tweets* alors que le match *LOSC Lille - Football Club Sochaux-Montbéliard* n'a été commenté que 374 fois.

4.1.2 Descriptions des résumés-minutes

Nous avons aussi récupéré les résumés minutes par minutes couvrant les matchs directement à partir des sites internet cités précédemment. Un tableau récapitulatif du nombre de minutes par résumés par match peut être vu à la table 4.2.

	Maximum	Minimum	Moyenne	Écart-type
Le Monde Sport	78	47	62	9.2
Football365	115	80	92	9.3
L'Équipe	94	55	67	10.7

TABLE 4.2 – Nombre de minutes par résumés-minutes

Malgré la plus grande qualité des résumés minutes-par-minutes par rapport aux *tweets* , il arrive que les résumés minutes-par-minutes comportent des erreurs et des approximations. Par exemple, nous avons sélectionné deux minutes décrivant la même action à la table 4.3. Ces deux minutes décrivent la même action, et sont en désaccord sur trois points :

- Le défenseur qui a contré le ballon. (Martins-Peireira pour le site Football365, Mathis pour L'Équipe).
- L'attaquant qui a tenté la frappe. (Alessandrini pour le site Football365, Oliveira, le portugais, pour L'Équipe).
- Le minutage de la minute. (74 contre 76)

Site	Temps	Minute
Football365	74	Bonne combinaison aux abords de la surface entre Alessandrini et Oliveira, mais Martins-Pereira est là pour mettre le pied au moment où Alessandrini tente la frappe et peut contrer le ballon.
L'Équipe	76	Après un double une-deux entre Alessandrini et Oliveira, le portugais tente sa chance à l'entrée de la surface, mais est contré par Mathis, qui s'est jeté dans ses pieds.

TABLE 4.3 – Nombre de minutes par résumés-minutes

Il est donc nécessaire de s'appuyer sur l'ensemble de ces résumés minutes-par-minutes, pour pallier les approximations de minutages et les erreurs des différents résumés. Nous ne pouvons pas nous fier au minutage utilisé par chaque site, chaque journaliste pouvant mettre plus ou moins de temps pour décrire une action. Pour effectuer cela, nous verrons à la section 5.1 une méthode pour synchroniser ces différents résumés minutes-par-minutes.

4.2 Environnement de développement

D'un point de vue plus technique, tous les développements au cours de ce stage ont été réalisés à l'aide du langage *Python* dans sa version 2.7, ce qui n'offre certes pas les meilleures performances tant d'un point de vue de temps d'exécution que de consommation mémoire, mais qui a l'avantage de permettre de produire des prototypes rapidement. Nous avons utilisé la bibliothèque *matplotlib* pour la génération des différents graphiques et diagrammes de ce rapport de stage, et plusieurs bibliothèques scientifiques telles que *NLTK*, *SciPy*. *NLTK* est une bibliothèque de référence dans le traitement automatique des langues, implémentant plusieurs techniques de références. *SciPy* est un ensemble de bibliothèques à usage scientifique.

Pour la récupération des *tweets* , nous nous sommes basés sur le travail d'un ancien stagiaire de l'équipe. Le logiciel utilise l'*API* Twitter afin de récupérer les *tweets* selon des conditions particulières (présence ou absence de *hastag*, date, etc.)

Nous récupérons les différents résumés minutes-par-minutes grâce à la libraire *beautiful soup* permettant de naviguer sur l'arborescence DOM des pages web et de récupérer ce qui nous intéresse.

Chapitre 5

Synchronisation des sources d'informations

Dans cette section, nous exposons les différentes synchronisations que nous avons effectuées entre les textes récupérés. Dans un premier temps nous décrivons la synchronisation entre les différents résumés minutes-par-minutes, puis nous présentons la synchronisation de cette fusion de résumés minutes-par-minutes avec les *tweets* . Le but est finalement d'obtenir des textes minutés que l'on puisse exploiter pour indexer la vidéo.

5.1 Fusion des résumés minutes-par-minutes

Comme vu à la section 4.1, nous avons à notre disposition des résumés minutes provenant de 3 sites différents. Nous souhaitons pouvoir fusionner ces différents résumés de manière automatique. Comme nous l'avons vu à la section 4.1.2, cette étape est nécessaire, car nous ne pouvons pas nous fier au minutage utilisé par chaque site. De plus, cette fusion permet d'obtenir plus de phrases pour une même action. Cela permet aussi de capturer certaines différences de vocabulaire (un résumé minute par minute peut utiliser l'expression "ouvrir le score" alors qu'un autre peut contenir "marquer un but"). Ces variations de vocabulaires seront utilisées quand nous alignerons ces minutes avec les *tweets* (voir section 5.2).

Nous souhaitons pouvoir faire correspondre tous les résumés minutes-par-minutes entre eux. Pour cela, nous proposons d'utiliser un algorithme classique d'alignement de séquences, la déformation temporelle dynamique. Cette méthode est présentée de manière un peu plus détaillée à la section suivante.

5.1.1 Déformation temporelle dynamique

L'algorithme de déformation temporelle dynamique (Dynamic Time Warping ou DTW) permet d'aligner deux séquences temporelles en fonction d'une fonction de coût, développé initialement pour

la reconnaissance automatique de parole [Sakoe and Chiba, 1978]. Une illustration est disponible à la figure 5.1

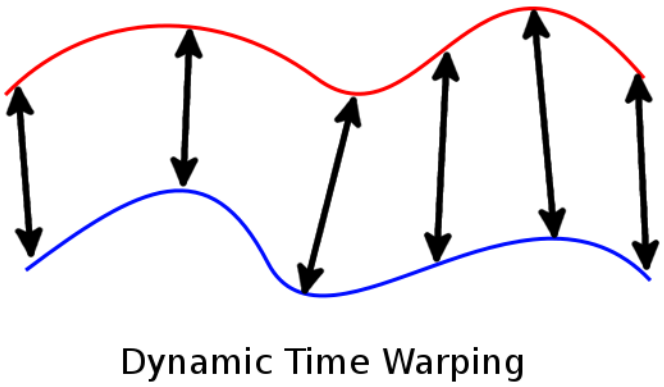

FIGURE 5.1 – Principe de la déformation temporelle dynamique

Nous considérons deux séquences de vecteurs A et B telles que $A = a_1, a_2, a_3, ..., a_i, ..., a_n$ et $B = b_1, b_2, b_3, ..., b_j, ..., b_n$. Les deux séquences peuvent être plaquées sur deux cotés d'une matrice comme dans la figure 5.2. L'algorithme se charge de trouver le chemin, déformant ainsi les séries temporelles, représenté par les points rouges sur la figure, qui minimise la fonction de coût (distance) entre les deux séquences temporelles.

L'algorithme DTW s'appuie sur de nombreuses hypothèses :

- Monotonie : le chemin trouvé est monotone, c'est-à-dire que les index i et j du chemin trouvé ne peuvent qu'augmenter ou rester égaux, et ne jamais diminuer.
- Continuité : le chemin trouvé est continu, c'est-à-dire que les index i et j du chemin trouvé ne peuvent qu'augmenter de 1 ou rester égaux.
- Limites : le chemin trouvé est borné. Les premiers éléments des deux séquences sont alignés, tout comme les derniers éléments des deux séquences. Cela implique que l'une des séquences ne peut pas être alignée à seulement une sous-séquence de l'autre.

L'algorithme est basé sur la programmation dynamique et est constitué de deux phases. La première phase consiste à créer une matrice de coût cumulé, en fonction de la fonction de coût. Cet algorithme est visible à l'algorithme 1. La complexité de cet algorithme est en $o(n \times m)$ avec n et m les longueurs respectives des deux séquences. Il est cependant possible de réduire l'espace de recherche, en ne s'intéressant qu'aux cases près de la diagonale, afin d'accélérer les calculs.

La deuxième phase de la déformation temporelle dynamique consiste à utiliser le retour sur trace (*backtracking*) pour trouver le chemin optimal de manière gloutonne, tel que décrit à l'algorithme

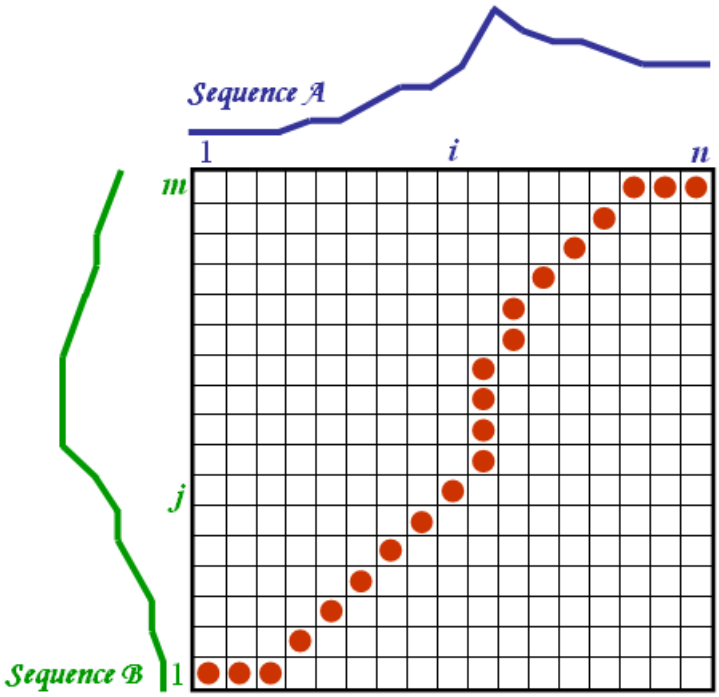

FIGURE 5.2 – Chemin représentant l'alignement entre deux séquences temporelles. **Source :** http ://www.psb.ugent.be/cbd/papers/gentxwarper/DTWalgorithm.htm

2.

Entrées : A une séquence temporelle
B une séquence temporelle
c une fonction de coût (distance)
Résultat : Dtw une matrice de coût accumulée
n $\leftarrow |A|$
m $\leftarrow |B|$
$dtw[] \leftarrow new[n \times m]$
dtw(0, 0) $\leftarrow 0$
pour $i = 1; i \leq n; i++$ **faire**
    $dtw(i, 1) \leftarrow dtw(i-1, 1) + c(i, 1)$
fin
pour $j = 1; j \leq n; j++$ **faire**
    $dtw(1, j) \leftarrow dtw(1, j-1) + c(1, j)$
fin
pour $i = 1; i \leq n; i++$ **faire**
    pour $j = 1; j \leq n; j++$ **faire**
        $dtw(1, j) \leftarrow c(1, j) + min\{dtw(i-1, j), dtw(i, j-1), dtw(i-1, j-1)\}$
    fin
fin

Algorithme 1: Construction de la matrice de coût accumulée

Entrées : Dtw une matrice de cout accumulée
Résultat : path chemin optimal (Alignement des séquences)
path [] ← new array
i = rows(dtw)
j = columns(dtw)
tant que *(i > 1) & (j > 1)* **faire**
    si $i == 1$ **alors**
        $j = j - 1$

    sinon si $j == 1$ **alors**
        $i = i - 1$

    sinon
        si $dtw(i-1,j) \leftarrow c(1,j) + min\{dtw(i-1,j), dtw(i,j-1), dtw(i-1,j-1)\}$ **alors**
            $i = i - 1$

        sinon si $dtw(i,j-1) \leftarrow c(1,j) + min\{dtw(i-1,j), dtw(i,j-1), dtw(i-1,j-1)\}$ **alors**
            $j = j - 1$

        sinon
            $i = i - 1$
            $j = j - 1$
        fin
        path.add((i,j))
    fin
fin

Algorithme 2: Chemin de déformation optimal

Nous avons vu que la déformation temporelle dynamique s'appuie sur une fonction de coût pour évaluer la distance entre les éléments de chaque séquence. Nous présentons donc à la section suivante la fonction de coût que nous avons proposée dans le cas d'informations textuelles.

5.1.2 Fonction de coût pour le textuel

Nous avons besoin d'une fonction de coût entre deux textes. Cette fonction doit représenter la distance entre le contenu de ces deux textes. Nous nous tournons vers le domaine de la recherche d'information, qui depuis quelques décennies s'intéresse à rechercher une information pertinente à travers une collection de documents. Nous utilisons donc l'approche vectorielle qui est utilisée en recherche d'information.

Représentation du texte en vecteurs

En premier lieu, chaque texte est décrit sous la forme de vecteurs. Cette représentation, la représentation sac de mots (*bag of words*), est très utilisée dans le domaine de la recherche d'information

et dans le traitement du langage naturel en général.

On suppose disposer d'un dictionnaire de l'ensemble des mots contenus dans tous les documents (en pratique, ce dictionnaire est obtenu en parcourant tous les documents du corpus). Chaque document est représenté par un vecteur de la même taille que le dictionnaire, dont la composante i indique le nombre d'occurrences du i-ème mot du dictionnaire dans le document. On remarque que la taille du dictionnaire influe sur le système. Pour cette raison, nous appliquons plusieurs traitements aux textes.

Un des premiers traitements que nous appliquons au texte est la suppression des mots vides. Les mots vides sont les mots les plus communs qui n'apportent aucune information utile pour notre problème. Par exemple, des mots comme les déterminants ou les auxiliaires sont des mots vides.

Nous avons aussi remplacé tous les liens présents dans les *tweets* par un mot-clé symbolique représentant uniquement les liens. Nous n'avons pas supprimé les liens car nous avons remarqué que beaucoup des *tweets* comportant des liens décrivaient les actions importantes du jeu.

Pondération

La méthode TF-IDF (*Term Frequency-Inverse Document Frequency*) est une méthode de pondération permet d'évaluer l'importance d'un terme contenu dans une requête par rapport à un document, relativement à un ensemble de documents ou corpus. C'est une méthode très classique en recherche d'information et en fouille de textes.

La mesure TF-IDF prend donc en compte la présence du terme requête dans le document, mais aussi de la popularité de ce terme dans l'ensemble de documents. En effet, la fréquence inverse de document vise à accorder un poids plus important aux termes les moins fréquents dans le corpus, car ces termes sont les plus discriminants.

Formellement, pour un terme t et un document d_i appartenant à un ensemble de documents $D = d_1, d_2, ..., d_i, ..., d_n$, la mesure tf-idf est calculée grâce à la formule suivante :

$$tf - idf_{t,i} = tf(t,i) * log\frac{n}{\{|d \in D|t \in d|\}}$$

$tf(t,i)$ étant la fréquence du terme t dans le document i et $\log \frac{n}{\{|d \in D|t \in d|\}}$ est la fréquence documentaire inverse, qui consiste à calculer le logarithme de l'inverse de la proportion de documents dans le corpus qui contiennent le terme t.

Similarité-Cosinus

Le modèle de similarité cosinus est un modèle vectoriel permettant d'introduire une mesure de similarité entre deux documents. La mesure de similarité cosinus entre deux documents d_i et d_j

représenté par deux vecteurs A et B se présentent sous la forme suivante :

$$cos(d_i, d_j) = \frac{A \cdot B}{||A|| \times ||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

$A \cdot B$ représente le produit scalaire des vecteur A et B et $||A||$ représente la norme euclidienne du vecteur A.

5.1.3 Modalités d'évaluation

Dans le cadre de l'évaluation (voir section 5.1.4), nous avons besoin de constituer une vérité-terrain quant à l'alignement des différents fichiers de minutes. En effet, nous devons comparer le résultat produit par le système automatique par rapport à un être humain. Nous alignons chacune de ces minutes manuellement. Pour ce faire, nous avons développé un petit script facilitant cette action. Chaque personne ne voit que deux minutes en même temps et doit uniquement dire si l'action représentée dans chacune de deux minutes correspond. Une capture d'écran de ce logiciel est visible à la figure 5.3. Nous avons fait aligner manuellement ces données par trois personnes, appelées annotateurs. L'alignement manuel des minutes étant une tâche consommatrice de temps malgré l'aide logicielle créée, nous avons fait le choix de faire cet alignement manuel que sur un seul match.

La table 5.1 contient le nombre de minutes alignées par annotateur et par fichier de résumé minutes par minutes.

Annotateur	Résumés 1 et 2	Résumés 1 et 3	Résumés 2 et 3
Annotateur 1 (Charles)	39	32	41
Annotateur 2 (Laurent)	40	31	40
Annotateur 3 (Vincent)	49	37	51

TABLE 5.1 – Nombres de minutes alignées par annotateur et par résumé-minutes

Accords inter-annotateurs

Nous nous sommes intéressés ensuite à l'accord inter-annotateurs. Les calculs d'accord inter-annotateurs permettent non seulement d'identifier la qualité des annotations produites (i.e. plus les annotateurs sont à l'unisson, meilleure est la qualité d'annotation), mais aussi à établir une limite supérieure quant aux performances maximales que l'on peut attendre d'un système automatique [Fort et al., 2012]. En effet, il est évident qu'il n'est pas raisonnable d'attendre d'un système automatique produise un résultat très pertinent sur une tâche pour laquelle différents annotateurs humains sont en désaccord. De plus, ces accords inter-annotateurs permettent donc d'évaluer indirectement la difficulté de la tâche effectuée.

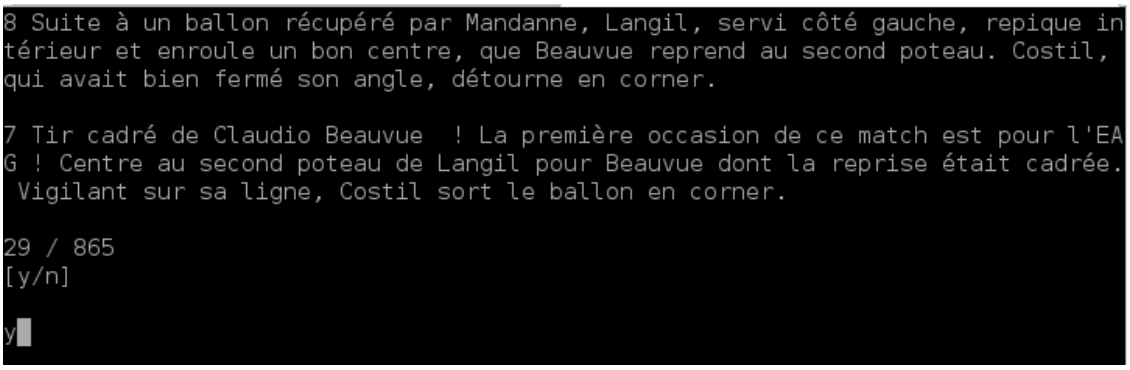

FIGURE 5.3 – Interface d'annotation

Pour mesurer ces accords inter-annotateurs, nous utilisons les mesures d'accords par corrélation κ de [Cohen, 1960] et π de [Scott, 1955]. Ces mesures sont soumises à plusieurs interprétations. Nous retiendrons l'interprétation de [Krippendorff, 1980]. L'accord inter-annotateurs est considéré comme "bon" quand κ est supérieur à **0.8**. Les résultats peuvent être consultés à la table 5.2.

	κ de Cohen	π de Scott
A1 et A2	0.841	0.841
A1 et A3	0.850	0.850
A2 et A3	0.854	0.853

TABLE 5.2 – Accords inter-annotateurs et intra-annotateurs

L'accord inter-annotateur est donc bon pour la tâche considérée. On notera tout de même que la valeur de κ est assez proche de la limite inférieure donnée dans l'interprétation de [Krippendorff, 1980].

5.1.4 Résultats

Nous évaluons donc notre fusion de minutes automatique par rapport à notre vérité-terrain. Nous utilisons des métriques standard dans le cadre de l'évaluation de processus automatiques, à savoir le rappel, la précision et la f-mesure.

Pour l'évaluation de notre fusion de minutes, le rappel est égal au nombre d'alignements trouvés par le système divisé par le nombre nombres d'alignements total dans la vérité-terrain. Le rappel met en avant la capacité du système à refuser les alignements de minutes incorrects.

Dans le cas de notre problème, la précision est définie par le nombre d'alignements corrects trouvés par le système automatique rapporté par le nombre d'alignements total trouvé par le système. La

précision mesure la capacité du système à donner tous les alignements de minutes corrects.

La F-mesure est une mesure qui prend en compte précision et rappel. Prendre en compte les deux mesures précédentes permet de montrer la capacité du système à refuser les alignements incorrects tout comme son aptitude à sélectionner tous les alignements corrects. La formule de la F-mesure est la moyenne harmonique du rappel et de la précision :

$$F = \frac{2 \times (P \times R)}{(P + R)}$$

La condition aux limites de l'algorithme DTW (section 5.1.1) oblige l'alignement de toutes les minutes de chaque résumé. Nous remarquons que les résumés minutes-par-minutes ne décrivent pas tout le temps la même action, nous décidons donc de filtrer le nombre de résultats. Nous appliquons un seuil minimum du score de similarité entre deux minutes alignées par l'algorithme DTW. Cela permet de ne retenir que les paires de minutes fortement proches. Les autres seront considérées comme décrivant d'autres actions non décrites par ailleurs.

La figure 5.4 comporte les résultats de cette fusion minutes par minutes.

Comme nous pouvons nous y attendre, le rappel et la précisions ont des évolutions antagonistes en fonction du seuil. La F-mesure est relativement stable aux alentours de **0.72**.

Il est assez difficile de replacer nos résultats par rapport à l'état de l'art. En effet, la plupart des alignements textuels sont réalisés entre des textes de langues différentes et reposent sur des hypothèses très différentes, et les quelques alignements monolingues de textes ont été réalisés entre deux niveaux de langues différentes de la même langue. Toutefois, notre F-mesure est un petit peu inférieur à l'état de l'art (**0.72** contre ≈ 0.8 [Bott and Saggion, 2011]). Nous expliquons cette différence par la taille des minutes alignées et la condition aux limites de l'algorithme DTW.

5.2 Récupération des informations en fonction de la popularité

5.2.1 Recherche des minutes en fonction de la popularité

Une fois la fusion des minutes effectuée, nous nous cherchons à extraire de cette fusion les informations les intéressantes pour notre problème. Nous supposons que les moments les plus intéressants d'un match sont aussi les moments les plus populaires, et donc les plus commentés par les utilisateurs de Twitter. Nous allons donc trouver les minutes des résumés les plus populaires. Pour ce faire, nous souhaitons utiliser les *tweets* comme indicateur de popularité, en alignant l'ensemble des *tweets* avec la fusion de minutes. De la même façon que pour la fusion des minutes, nous ne pouvons pas nous baser sur l'horodatage des *tweets* , les auteurs de *tweets* commentant à des moments différents. La correspondance minutes/*tweets* peut nous permettre d'obtenir, à travers le débit des *tweets* , les minutes les plus importantes, ainsi que celles ayant marquées le public[Lanagan and Smeaton, 2011].

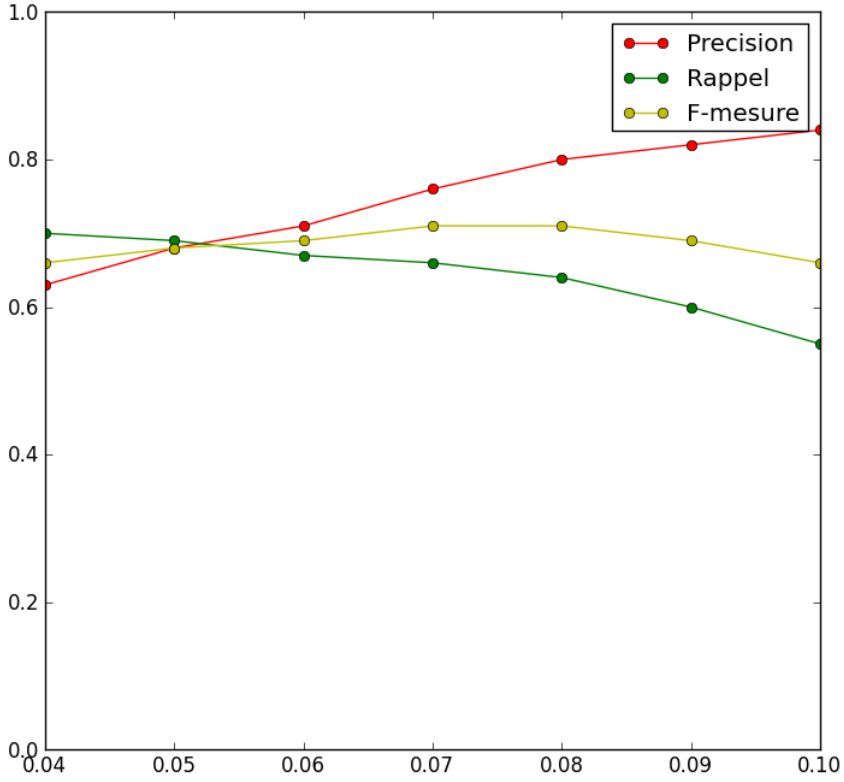

FIGURE 5.4 – Résultats de la fusion de minutes en fonction du seuil

Pour remédier à ce problème, nous proposons d'aligner cette fusion de minutes aux *tweets* de la même façon que pour la création de fusion de minutes. Pour palier le problème de la taille des *tweets* (voir section 3.2.1), nous créons des groupes arbitraires de *tweets* , situés dans une fenêtre de 30 secondes, et nous cherchons à aligner ces groupes de *tweets* avec les différentes minutes ou fusion de minutes. Nous sommes conscients que cela peut mener à grouper des *tweets* évoquant des actions différentes, mais nous supposons que ce phénomène reste marginal dans la quantité de *tweets* traités.

Nous utilisons donc l'algorithme DTW avec la même fonction de coût que pour la fusion des minutes (section 5.1.1 et section 5.1.2). À cause de la condition aux limites de l'algorithme DTW, nous sommes obligés d'appliquer un seuil pour filtrer le nombre minutes intéressantes pour notre problème. Nous appliquons un seuil basé sur un nombre minimal de *tweets* associés pour obtenir les minutes les plus populaires. Ce seuil a pour valeur la moyenne entre le plus petit nombre de *tweets*

par fenêtre de 30 secondes et le plus grand nombre de *tweets* par fenêtre. Nous extrayons de ces minutes les mots les plus importants afin de créer notre nuage de mot illustré en figure 2.1 de la section 2.

Nous utilisons aussi le reconnaisseur d'entités nommées *nero*[1], créé par l'équipe à ces minutes, afin de capturer les acteurs impliqués dans l'action. Ces différents acteurs se retrouveront dans le nuage de mots.

5.2.2 Résultats

Nous présentons dans cette sous-section les résultats de l'alignement des *tweets* et des résumés minutes-par-minutes selon deux méthodologies différentes.

Première évaluation naïve

Comme [Lanagan and Smeaton, 2011], nous évaluons notre technique selon le nombre d'évènements important récupérés. Les évènements que nous qualifierons d'important seront les buts. Nous comptons pour chaque match de football trois éléments :

- Les buts trouvés par notre système.
- Les buts manqués par notre système.
- Les faux positifs : des minutes ne décrivant pas des buts et qui ont été sélectionnées par notre système.

La figure 5.5 montre le nombre des buts récupérés par cette méthode.

Nous pouvons voir que nous capturons la plupart des buts de tous les matchs, car ces actions sont très commentées. En revanche, nous capturons aussi beaucoup de minutes qui ne décrivent pas de buts, mais ce sont principalement sur des matchs très commentés comme *Lyon-Paris* et *Paris-Marseille*, qui sont des matchs très attendus des spectateurs de football. Le nombre de ces fausses minutes est peut-être dû au seuil qui peut être affiné.

Évaluation fine

À cause des mêmes raisons que précédemment (voir section 5.1.3), nous n'avons constitué la vérité-terrain que pour un seul match. Nous avons donc aligné manuellement les groupes de *tweets* de 30 secondes avec les résumés minutes-par-minutes. Cet alignement manuel a été réalisé en respectant les différentes conditions de l'algorithme DTW : monotonie, continuité, et conditions aux limites. Nous évaluons donc l'alignement de *tweets* ainsi réalisé. Pour cela, pour chaque *tweet*, nous regardons s'il est aligné avec la même minute que la vérité-terrain. Nous utilisons utilisons uniquement la précision comme métrique par rapport à la section 5.1.4, le rappel n'ayant pas de sens, toutes les minutes étant alignées. Les résultats sont visibles au tableau 5.3.

1. http ://nero.irisa.fr

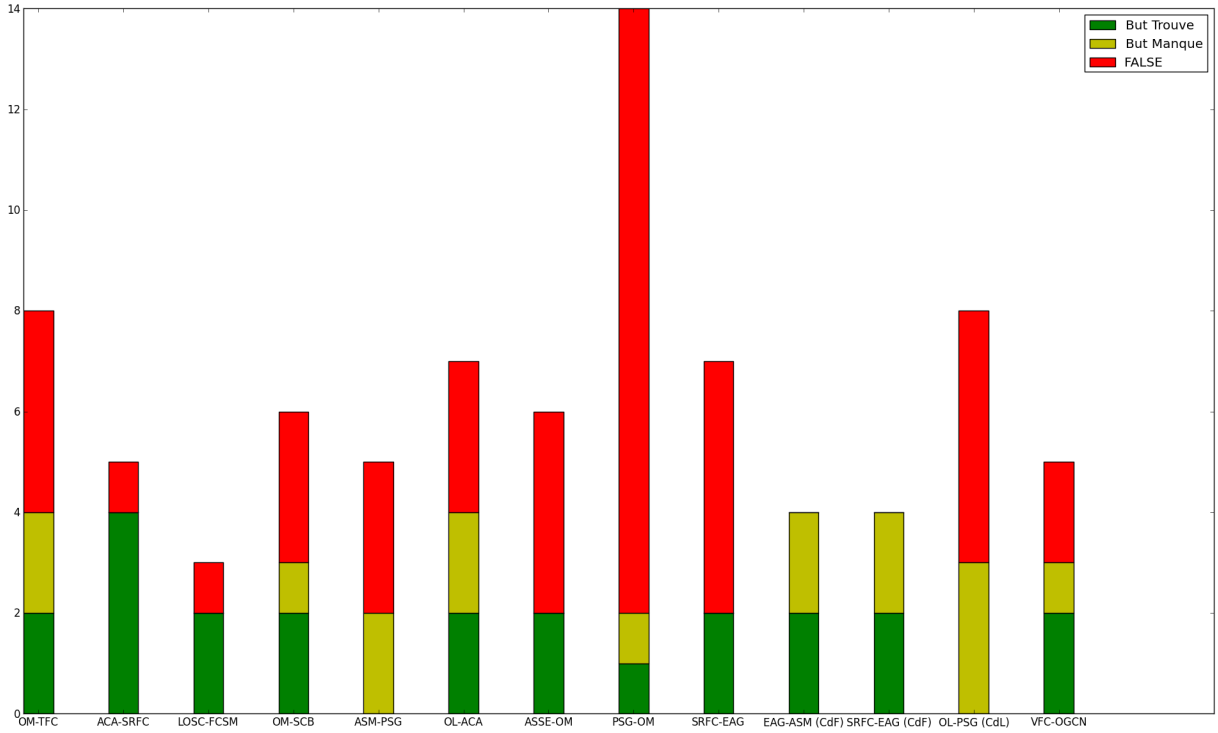

FIGURE 5.5 – Résultats de la récupération de minutes en fonction de la popularité

	Précision
Résultats	0.783

TABLE 5.3 – Résultats de l'alignement des minutes avec les *tweets*

Chapitre 6

Prise en compte de la polarité

Nous nous sommes intéressés également à la polarité des *tweets* . La polarité, ou l'opinion, représente le point de vue de l'auteur d'un texte vis-à-vis du sujet de ce texte, et comporte donc une interprétation subjective (voir [Pang et al., 2002] pour une définition plus formelle). Dans notre cas, cette polarité est définie simplement comme l'appartenance aux supporters de l'une ou l'autre équipe du match considéré. L'intérêt de cette étude est de vérifier si les matchs sont perçus différemment d'un camp à l'autre. Cela pourrais permettre de proposer des traitement différenciés selon les affinités de l'utilisateur.

6.1 But de l'expérience

Les *tweets* étant, de par leur nature de microblogs, propices à l'expression d'opinions, nous nous sommes intéressés à la polarité des *tweets* afin d'étudier l'influence de la celle-ci sur le débit de *tweets* au cours d'un match. Nous pensons, grâce à cette étude de la polarité des *tweets* , pouvoir créer des descriptions des matchs orientés en fonction d'une équipe ou de l'autre. Nous nous sommes aussi intéressés au débit des *tweets* en fonction de la polarité. En effet, on peut s'attendre intuitivement à un nombre de *tweets* différents selon l'action commentée, en particulier de l'équipe responsable de cette action.

6.2 Annotation des ensembles de données

Nous avons donc annoté manuellement l'ensemble des *tweets* de plusieurs matchs en trois catégories :

- Dom : l'auteur soutient l'équipe jouant à domicile
- Ext : l'auteur soutient l'équipe jouant à l'extérieur
- Neutre : L'auteur ne soutient aucune des deux équipes

La tâche d'annotation étant consommatrice de temps et assez fastidieuse, nous n'avons pu nous constituer qu'un volume restreint de données. Nous avons annoté seulement deux matchs de foot-

ball de la Ligue 1 : le match Lille - Sochaux et le match Ajaccio - Rennes. Les résultats que nous présentons ici ne sont peut-être pas représentatifs de l'ensemble des matchs de football, mais fournissent des enseignements importants pour notre objectif.

6.3 Résultats

Nous n'obtenons pas de différences significatives dans les débits des *tweets* en fonction de la polarité. Les pics de chaque courbe se situent au même moment, tout comme les périodes creuses. Des graphiques représentant le débit de *tweets* selon les plusieurs polarités sont visibles aux figures 6.1 et 6.2. Nous avons fait un test-t de Student appairé sur la différence entre deux périodes de temps consécutives pour chacune des deux équipes. Notre hypothèse nulle est qu'il existe une différence entre les deux équipes. Les résultats peuvent être consultés au tableau 6.1. Nous pouvons donc rejeter l'hypothèse nulle, il n'existe pas de différences statistiquement significatives entre les deux série avec une probabilité de 94% et 98% respectivement sur les matchs de football Lille - Sochaux et Ajaccio - Rennes.

Les actions de chacune des équipes sont commentées au même moment, en bien ou en mal. La plupart des auteurs de *tweets* commentent chaque action, indépendamment de l'équipe à l'origine de cette action. Si l'équipe qu'ils supportent dominent, ils se mettent à l'encourager, alors qu'ils commentent les erreurs de cette même équipe dans le cas contraire.

	t-value	p-value
Lille - Sochaux	-0.07	0.94
Ajaccio - Rennes	-0.02	0.98

TABLE 6.1 – Résultats du test-t de Student sur la différence de *tweets* entre périodes de temps consécutives

La prise en compte de la polarité pourrais nécéssiter des traitements plus fins, s'appuyant non pas sur le nombre de *tweets* mais sur une analyse du contenu de ces *tweets* . Plusieurs pistes évoquées en section 3.1.4 pourraient être suvies pour ce faire.

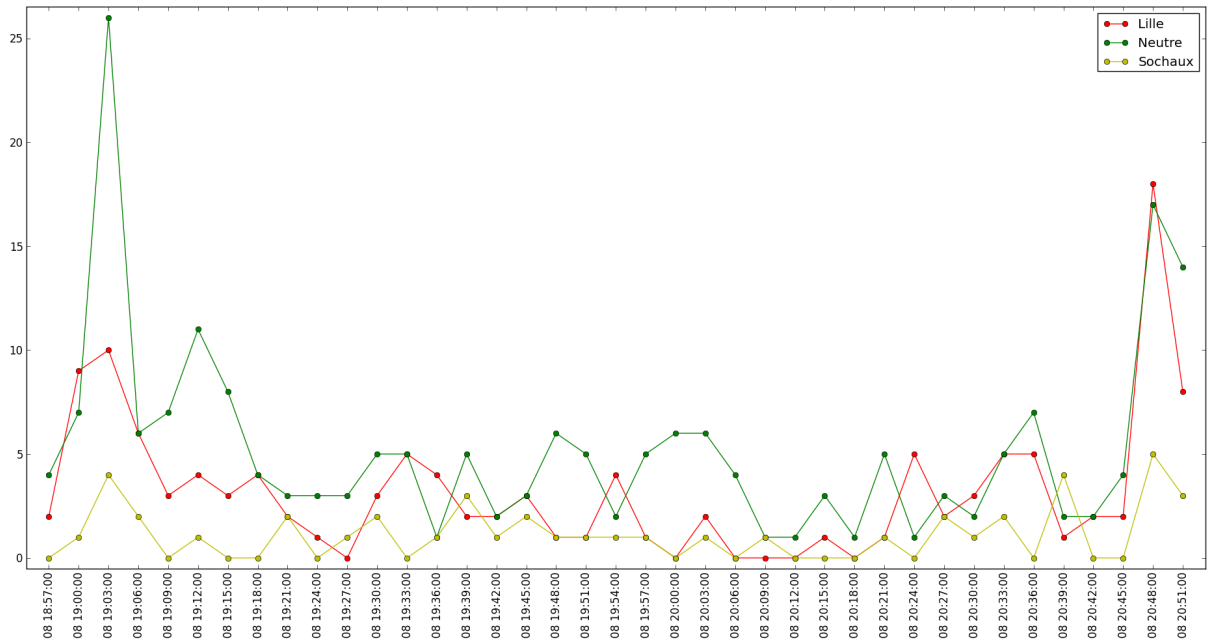

FIGURE 6.1 – Débits de *tweets* pendant le match Lille-Sochaux en fonction de la polarité

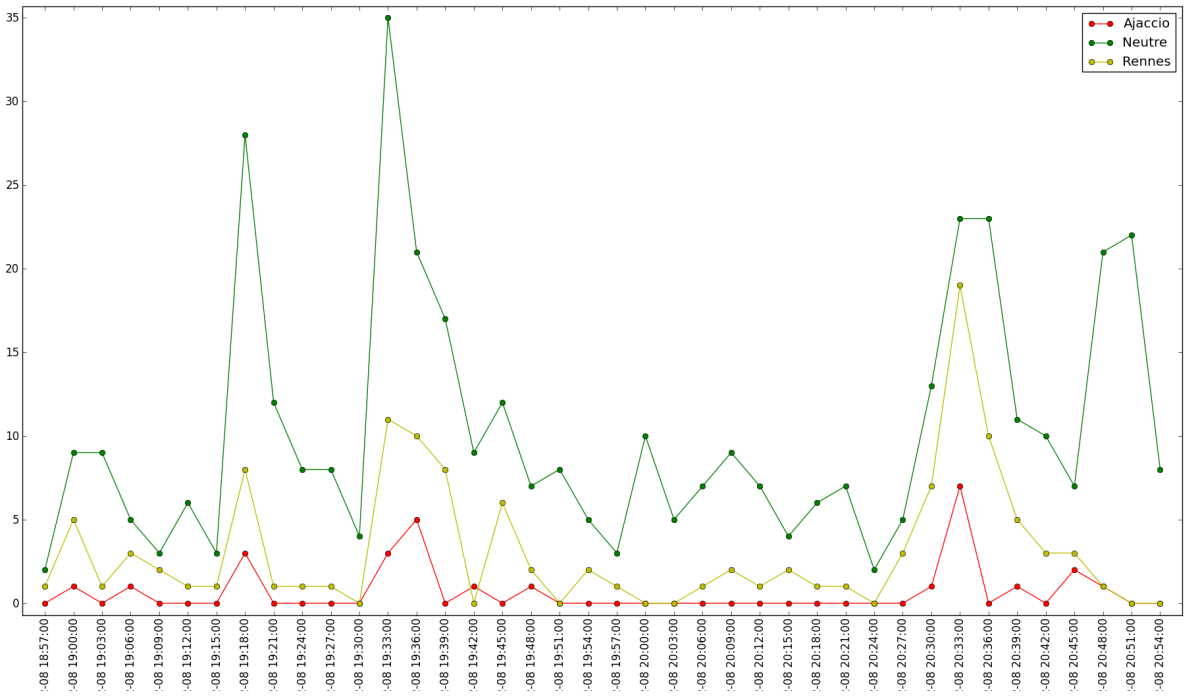

FIGURE 6.2 – Débits de *tweets* pendant le match Ajaccio-Rennes en fonction de la polarité

Chapitre 7

Pistes d'amélioration

Dans ce chapitre nous revenons sur plusieurs pistes qu'il nous semble intéressant d'explorer à la suite de ce stage.

De nombreuses améliorations sont possibles sur l'extraction d'information à partir de *tweets* . Cependant, la plupart d'entre elles nécessitent d'utiliser des techniques d'apprentissage supervisé, et donc une bonne vérité terrain dont la constitution est difficile, qui se trouve être un gros problème pour sa constitution. L'annotation est en effet très fastidieuse, de part le nombre de *tweets* par match de football, ainsi que de part le contenu de ces microblogs, qui dérivent assez facilement du sujet initial.

Cepoendant, sur la base de nos travaux d'alignements, nous pourrions essayer d'apprendre automatiquement des correspondances de vocabulaire à l'aide d'apprentissage supervisé à partir des alignements entre *tweets* et résumés minutes-par-minutes effectués. Cela permetterait par la suite de faire de la classification des *tweets* directement en actions, sans recours aux résumés minutes-par-minutes. Par exemple, le système pourrait apprendre que les signes de ponctuation " ! ! !" dans les *tweets* correspondent avec une certaine probabilité à un but dans le match.

Il serait intéressant également d'entraîner nous même un reconnaisseur d'entités nommés directement sur les *tweets* , et éviter d'utiliser le reconnaisseur déjà entraîné créé par l'équipe pour avoir un outil vraiment adapté aux *tweets* . Cela nous aurait permis de passer outre les étapes d'alignement entre les résumés minutes-par-minutes et les *tweets* .

Une des pistes d'amélioration évidente à long terme de ce stage est l'application de ces principes sur des événements différents que des matchs de football. Des projets sur l'application de ces méthodes aux journaux TV et émissions de reportage sont à l'étude.

Chapitre 8

Conclusion

Au cours de ce stage, nous utilisons ensemble des techniques de différentes problématique de recherche (alignement de séquences, recherche d'information) afin d'aligner plusieurs documents décrivant le même match de football dont nous extrayons les parties les plus intéressantes pour l'indexation.

Nous utilisons les avantages des deux sources d'information afin d'obtenir des informations de qualité tout en étant intéressantes d'un point de vue humain pour l'indexation. De plus, ces différents alignements nous permettent de contourner certaines faiblesses de ces textes. La fusion des résumés minutes-par-minutes nous permet de récupérer des variations morpho-syntaxiques, tandis que l'alignement de cette fusion aux flux des *tweets* nous permet de sélectionner les informations utiles pour l'indexation.

Dans ce rapport de stage, nous avons proposé une solution pour indexer des matchs de football avec des informations d'un assez haut niveau sémantique à l'aide de *tweets* et de texte journalistique. Dans un premier temps, nous avons commencé ce rapport en posant les bases de notre problème. Puis, nous avons vu qu'il existe de nombreuses techniques pour extraire de l'information à partir de textes dans l'état de l'art, mais que ces techniques ne sont pas toujours adaptées aux microblogs, qui possèdent des caractéristiques singulières comme la longueur et le coté informel du texte. Ensuite, nous avons détaillé la constitution de notre corpus de données, constitué de deux sources d'informations différentes et complémentaires. Puis nous avons montré comment combiner les avantages de chacune de ces sources d'informations afin d'indexer les matchs de football. Nous nous sommes intéressés au passage à la fusion de résumés minutes-par-minutes ainsi qu'à l'accord inter-annotateurs. Enfin, nous nous sommes intéressés à la polarité dans les messages. Dans le dernier chapitre, nous expliquons des pistes d'amélioration de notre travail.

Un des aspects à ne pas négliger lors de d'études sur des données est la collecte de ces données ainsi que la constitution d'une vérité-terrain solide. C'est pourquoi nous nous sommes intéressés à la problèmatique de l'accord inter-annotateurs lors de cette création de vérité-terrain.

Bibliographie

[Arnulphy, 2012] Arnulphy, B. (2012). *Désignations nominales des événements : étude et extraction automatique dans les textes*. Thèse de doctorat, Université Paris Sud-Paris XI.

[Baghdadi, 2010] Baghdadi, S. (2010). *Extraction multimodale de métadonnées de séquences video dans un cadre bayésien*. Thèse de doctorat, Université Rennes 1.

[Barnard et al., 2003] Barnard, M., Odobez, J.-M., and Bengio, S. (2003). Multi-modal audio-visual event recognition for football analysis. In *Neural Networks for Signal Processing, 2003. NNSP'03. 2003 IEEE 13th Workshop on*, pages 469–478. IEEE.

[Bott and Saggion, 2011] Bott, S. and Saggion, H. (2011). An unsupervised alignment algorithm for text simplification corpus construction. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 20–26. Association for Computational Linguistics.

[Cohen, 1960] Cohen, J. (1960). A coefficient of agreement for nominal scales. In *Educational and Psychological Measurement*, pages 27–46.

[Daille, 2002] Daille, B. (2002). Découvertes linguistiques en corpus, mémoire d'habilitation à diriger des recherches en informatique. *Université de Nantes*.

[Deveaud et al., 2013] Deveaud, R., Boudin, F., et al. (2013). Contextualisation automatique de tweets à partir de wikipédia. In *Actes de la conférence CORIA 2013*.

[Fort et al., 2012] Fort, K., Claveau, V., et al. (2012). Annotation manuelle de matchs de foot : Oh la la la ! l'accord inter-annotateurs ! et c'est le but ! In *Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 2 : TALN*, pages 383–390.

[Gimpel et al., 2010] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2010). Part-of-speech tagging for twitter : Annotation, features, and experiments. Technical report, DTIC Document.

[Krippendorff, 1980] Krippendorff, K. (1980). *Content Analysis : An Introduction to Its Methodology*. Sage Publications, London.

[Krupka and Hausman, 1998] Krupka, G. R. and Hausman, K. (1998). Isoquest inc. : Description of the netowl (tm) extractor system as used for muc-7. In *Proceedings of MUC*, volume 7.

[Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data.

[Lanagan and Smeaton, 2011] Lanagan, J. and Smeaton, A. F. (2011). Using twitter to detect and tag important events in live sports. *Artificial Intelligence*, pages 542–545.

[Liu et al., 2011] Liu, X., Zhang, S., Wei, F., and Zhou, M. (2011). Recognizing named entities in tweets. In *ACL*, pages 359–367.

[McCallum and Li, 2003] McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics.

[Pang et al., 2002] Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up ? : sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

[Ritter et al., 2011] Ritter, A., Clark, S., Etzioni, O., et al. (2011). Named entity recognition in tweets : an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics.

[Sakoe and Chiba, 1978] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1) :43–49.

[Scott, 1955] Scott, W. A. (1955). Reliability of content analysis : The case of nominal scale coding. *Public opinion quarterly*.

[Sebastiani, 2002] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1) :1–47.

[Thelwall et al., 2010] Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12) :2544–2558.

[Yow et al., 1995] Yow, D., Yeo, B.-L., Yeung, M., and Liu, B. (1995). Analysis and presentation of soccer highlights from digital video. In *proc. ACCV*, volume 95, pages 11–20. Citeseer.