Deep Neural Architectures for Automatic Representation Learning from Multimedia Multimodal Data

Vedran Vukotic

Thèse

UNIVERSITE BRETAGNE LOIRE

THESE INSA Rennes
sous le sceau de l'Université Bretagne Loire
pour obtenir le titre de
DOCTEUR DE L'INSA RENNES
Spécialité : Informatique

présentée par
Vedran Vukotić

ECOLE DOCTORALE : *MATHSTIC*
LABORATOIRE : *IRISA - UMR6074*

Deep Neural Architectures for Automatic Representation Learning from Multimedia Multimodal Data

Thèse soutenue le 26.09.2017
devant le jury composé de :

Georges Quénot
Directeur de recherche au CNRS, LIG / Président
Martha Larson
Professeur à Radboud University / Rapporteuse
Tinne Tuytelaars
Professeur à KU Leuven / Rapporteuse
Benoit Huet
Professeur à EURECOM / Examinateur
Christian Raymond
Maitre de Conférences à l'INSA de Rennes / Co-encadrant
Guillaume Gravier
Directeur de recherche au CNRS, IRISA / Directeur de thèse

Deep Neural Architectures for Automatic Representation Learning from Multimedia Multimodal Data

Vedran Vukotić

En partenariat avec

Stay curious.

—Anonymous

Acknowledgements

I would like to thank my supervisors, Christian Raymond and Guillaume Gravier, for creating my PhD position, helping me find a grant while I was still an intern in the team, and for passionately guiding me through my research by providing numerous interesting problems to explore. The topic of my thesis could not have been a better fit and 3 years passed in a femtosecond while I was busy following my passions and thoughts. I will always remember my PhD as a very constructive and interesting period of my life and for that I have to thank you — my supervisors. In addition to your scientific knowledge and curiosity, I will also never forget the kindness and supportiveness you have shown. Thank you for everything!

Furthermore, I would like to thank the jury members and evaluators: Georges Quénot, Martha Larson, Tinne Tuytelaars and Benoit Huet for evaluating my work, providing me with valuable feedback and numerous new ideas, and for your kind words and interesting discussions during the defense.

Thank you Jan van Gemert and Silvia Laura Pintea for welcoming me at TU Delft and for making me feel at home for 4 months in Delft. On top of the interesting discussions we had, I will never forget how passionate everybody was in the lab: daily 5 minutes presentations, weekly talks, a reading group and regular internal poster sessions to exchange ideas and give or receive feedback.

I would like to thank all the former and current team members of LinkMedia at Irisa and the members of the Pattern Recognition & Bioinformatics Group at TU Delft for all the nice discussions and the time spent together. Each and every one of you helped me at some point either with practical issues or by suggesting conferences, workshops or grants. Collectively, your help was immense and I could not have done it without you! A big thank you to all the LinkMedia members for the awesome poster, the nice gifts and your kind wishes for my PhD defense!

In particular, I would like to thank: Anca, for introducing me to video hyperlinking, summer-school and visiting grants and for helping me multiple times at CIREFE when I had to rush home for my grandmother; Ricardo, for checking my studio when I was at TU Delft; for the numerous tech discussions we had and the nice biking trip in Brocéliande; Miaojing and Li for the numerous trips and voyages we did together and for the nice and interesting discussions. It was nice to spend time with you outside the lab!

When I arrived in Rennes, I knew I was going to explore different scientific directions and gain new insights and views. However, I would have never expected that I would rediscover my home island in Rennes thanks to the GSR (Groupe Spéléologique de Rennes). With you, I have discovered the beautiful underground of France but most importantly, you inspired me to start looking for hidden underground treasures on my island and now, after 30 years, I rediscovered it in a way I would have never imagined it: full of beautiful caves with pits, chambers, seawater lakes and with a multitude of amazing formations that took centuries to form. Like in science, you made me shine a light in a previously unknown path and discover a completely unexpected side of what I thought I knew well. Thank you for all the nice moments we spent together above and below ground!

Last but not least, I would like to thank the people that influenced my life choices. Thank you Siniša and Josip for transmitting me your passion for computer vision and deep learning. Without you I might have not ended up loving this topic in the right time! Thank you Davor for always inspiring me with more topics that we are able to follow and for being a great friend. A big thank you goes especially to my grandma Noyes and mother Arlen for motivating me to never stop asking questions, having a critical and analytic mindset and for teaching me the importance of curiosity and of always following our passions. I will never forget the enormous influence you had on me!

Contents

Chapter 0

Resumé

Contents

0.1 Contexte

Avec la récente résurgence des réseaux de neurones, le rapide essor des méthodes d'apprentissage profond et la prolifération de données massives non annotées, les algorithmes non supervisées ont gagné en popularité de part leur faculté à extraire de l'information depuis ces données [14]. Les méthodes modernes d'apprentissage profond ainsi que les récentes évolutions matérielles (GPU) permettent l'apprentissage de réseaux de neurones depuis des données quasiment brutes, c'est à dire sans la fastidieuse et coûteuse opération manuelle d'extraction de caractéristiques, jusqu'à la prédiction de la tâche finale. Au final ces réseaux apprennent de bout en bout à la fois une représentation des données ainsi que sa projection vers la prédiction sous jacente à la tâche finale facilitant ainsi leur déploiement. Un bon indicateur de la popularité et du succès de cess méthodes est le défi ImageNet où depuis 2013, quasiment tous les participants ont utilisés des méthodes neuronales profondes [92] et les seuls participants à avoir recours à une extraction de caractéristiques manuelle ont été relégués en queue du classement. L'apprentissage profond et donc l'apprentissage automatique de représentations n'est pas en vogue uniquement dans le domaine de la vision par ordinateur, d'autres domaines tels que le traitement des langues na-

turelles, la reconnaissance de la prole, les systèmes de recherche d'information multimodaux et bien d'autres suivent cette vague.

Les objectifs scientifiques ont évolué de l'extraction de caractéristiques et combinaison de classifieurs au développement d'architecture neuronales qui apprennent automatiquement des représentations et les exploitent efficacement pour une multitude de différentes tâches. Différents type d'architecture de réseaux profonds existent et chacune d'entre elles est dédiée à une classe de problème spécifique. En traitement des langues naturelles ou autre problème de modélisation de séquences les réseaux récurrents ont permis le traitement plus efficace de ces séquences de longueur variable. Le dernier modèle de ce type de réseau embarque des "portes" permettant d'apprendre quelle information mémoriser et quelle information oublier et défini actuellement l'état de l'art en compréhension de la parole [129], en traduction [4], en système de question/réponse [125] et différentes autres tâches dont les observations dépendent de séquences et où la modélisation et la sélection d'information dans cette séquence est primordiale. En vision par ordinateur, les réseaux convolutifs sont utilisés pour obtenir automatiquement de bonnes représentations d'images [101] et sont utilisés dans de nombreuses tâches du domaine de la vision par ordinateur telles que la classification [61], la segmentation sémantique [64], l'estimation de saillance [41] et bien d'autres. Lorsqu'ils sont transposés, ces réseaux sont souvent appelés réseaux "déconvolutifs" et sont utilisés pour la génération d'images synthétiques depuis une représentation d'image existante où une source de bruit aléatoire.

L'avantage majeur apparaît lorsqu'on combine différent types de réseau pour former des architectures complexes capables de traiter différentes données, en apprendre des représentations efficaces, les combiner ou les transformer pour être efficace sur différentes tâches. C'est l'aspect le plus intéressant en apprentissage non-supervisé où de tels réseaux peuvent être entraînés en exploitant les ressources considérables de données non-annotées accessible sur internet, sans nécessité de recourir à des processus coûteux et fastidieux d'annotation manuelle. Les exemples typiques inclus le traitement multimodal non-supervisé de vidéos non-annotées où les réseaux de neurones profonds sont capables de fusionner les informations obtenues depuis la transcription automatique de la parole et les représentations visuelles obtenues par un réseau convolutif dans le but d'apparier correctement des segments de vidéo. Cette tâche spécifique d'appariement est très en vogue dans les dernières campagnes d'évaluation internationales telles que MediaEval ou TRECVID. D'autres tâches d'apprentissage typique où sont impliqués différentes type d'architecture de réseau sont, mais ne sont pas limitées à, la prédiction de mouvement [121, 123], génération d'image à partir de texte [89, 133], la reconstruction d'images [131, 134] et bien d'autres.

Dans cette dissertation, nous évaluons la thèse que les plongements neuronaux (neural embedding) sont adaptés pour la fusion multimodale. Nous allons dans une première partie nous intéresser au développement d'architectures exploitant les informations textuelles ou visuelles de manière indépendante puis dans une second temps sur des architectures qui exploitent la combinaison des 2 sources d'informations. Nous évaluons différentes architectures récurrentes pour la compréhension de la parole et nous les comparons à l'état de l'art; nous proposons ensuite des architectures convolutives simples qui prédisent le mouvement depuis des images statiques, nous tentons de combler l'écart entre les méthodes d'apprentissages classiques et les réseaux neuronaux très profonds. En ce qui concerne les méthodes non-supervisées pour fusionner informations visuelles et textuelles, nous proposons différentes architectures qui sont capables de mieux exploiter de grosses collections

de vidéos et nous améliorons la recherche de contenu aussi bien en terme de performance qu'en terme de diversité de contenu retrouvé. Toutes les méthodes proposées sont évaluées sur des "benchmarks" tels que TRECVID et définissent maintenant un nouvel état de l'art. Nous évaluons également des méthodes permettant de mieux visualiser des modèles de réseaux de neurones.

0.2 Espace de représentation et apprentissage profond

Les avancées récentes en apprentissage profond ont changé la manière d'aborder les problèmes en traitement des langues; vision par ordinateur ou multimédia. L'extraction manuelle de caractéristiques a perdu de son importance avec la possibilité d'apprentissage automatique des représentations. Dans le traitement de textes, les représentations sacs de mots ont été remplacées par des méthodes qui apprennent des représentations de manière non-supervisée à partir de corpus [70, 59]. En vision par ordinateur ce phénomène est encore bien plus saillant et les caractéristiques extraites manuellement [65, 6, 69] sont remplacées par des représentations apprises automatiquement durant l'apprentissage supervisé du réseau sur une tâche spécifique [98]. Dans le domaine du multimédia la même tendance est suivie où les méthodes pour fusionner différentes modalités ont été développées [55, 73, 30, 15].

Dans le but d'utiliser des données provenant de différentes modalités (texte, images, etc.) avec des réseaux de neurones, il est nécessaire de projeter chaque entrée dans un espace de représentation continu. Cette projection, en plus d'être nécessaire pour les réseaux de neurones possède d'intéressante propriétés intrinsèques [70, 98, 85, 67]. Nous introduisons l'architecture neural de base que nous allons utiliser intensivemment dans ce travail qui permet l'apprentissage de représentations de textes et d'images, le passage de l'une à l'autre ainsi que la synthèse d'exemples artificiels dans le domaine original (e.g., espace des images ou textes) étant donnée une représentation.

Nous définissons une "modalité" comme une collection de données agrégées par un outil d'acquisition [55]. Deux outils d'acquisition typique sont: i) l'acquisition d'image (effectué typiquement avec des capteurs CMOS ou CCD) qui capture le monde et produit une représentation discrète dans l'espace des images (images) ou dans une espace temporel d'images (vidéos), et ii) l'acquisition du son (effectué généralement avec différents type de microphones and des convertisseurs analogique digitaux. Chaque modalité peut être transformée et représentée de plusieurs façons. Un signal audio contenant de la parole peut être automatiquement transcrit et exploité comme texte ou il peut être représenté au moyen d' i-vecteurs [24] et exploité pour la tâche de reconnaissance du locuteur. Les images clef d'une vidéo peuvent être décrites avec des concepts compréhensibles (mots) tels que ImageNet [92] ou avec des caractéristiques obtenues par des réseaux de neurones convolutifs [98, 51].

La figure 1 illustre deux modalités: audio et visuelle, et les trois différents niveaux auxquelles elles peuvent être représentées:

- **l'espace original** - c'est l'espace où les données sont représentées après l'acquisition et la numérisation. Cela peut être un signal 1D discret pour l'audio ou une séquence de mesures RVB pour une image.
- **l'espace conceptuel** - c'est une espace qui décrit l'espace original avec des concepts clefs ou mots clefs. Ce genre d'espace est en général peu adapté pour faire de

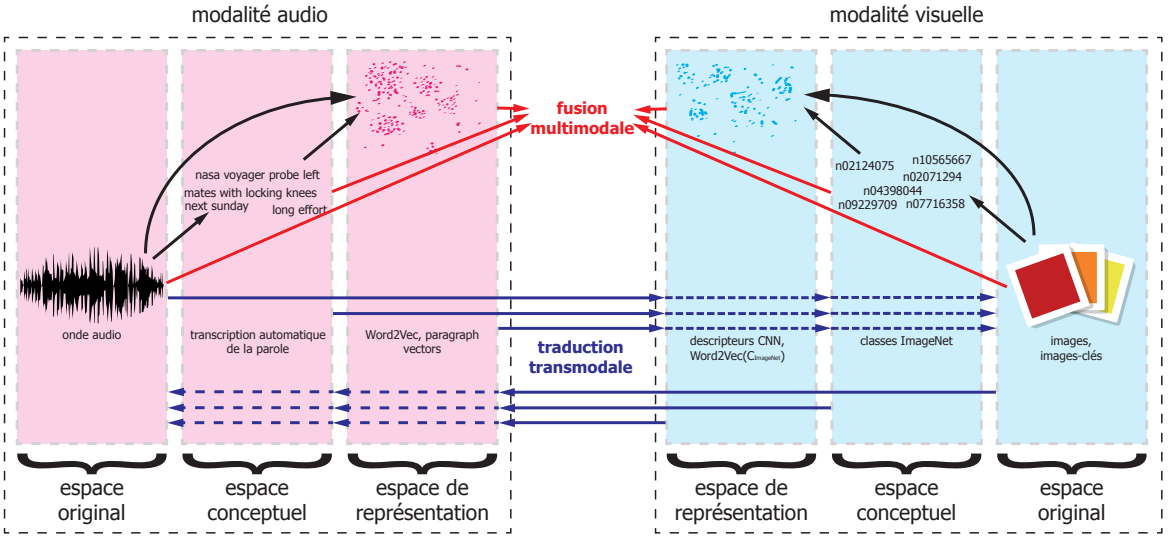


Figure 1: Deux modalités différentes (audio et visuelle), chacune présentée à différents niveaux. La combinaison de deux modalités à des niveaux arbitraires (fusion multimodales) est notée en rouge. Passer d'une modalité à une autre (traduction de modalité), à des niveaux arbitraires est noté en bleu.

l'apprentissage automatique mais il est en général interprétable humainement et fourni un résumé simple de ce qui a été acquis dans l'espace original. Les exemples typiques sont les transcriptions automatique de la parole depuis une source audio ou simplement l'extraction des principaux mots clefs et les concepts ImageNet décrivant le contenu de la trame clef d'une vidéo.

- **l'espace de représentation** - c'est l'espace le plus utile pour faire de l'apprentissage automatique. Il peut être un espace de représentation discret obtenu depuis l'espace conceptuel (e.g., sac de mots obtenues à partir de transcriptions automatique) ou un espace de représentation continu (e.g. word2vec). Dans les applications ayant recours à l'apprentissage profond au moyen de réseaux de neurones, les représentations continues sont en général utilisées et produisent des performances état de l'art. Les espaces de représentation, particulièrement les espaces de représentations continus sont moins interprétables humainement que les espaces conceptuels mais sont très pertinent pour les application d'apprentissage automatique et proposent des propriétés intéressantes [70]. Ces intéressantes propriétés des représentations continues peuvent également être observées dans le domaine orignal par synthèse grâce aux réseaux génératifs adverses [35, 85, 43].

Quand on travaille avec des données qui contiennent plusieurs modalités (e.g. images sous-titrées, vidéos, transcriptions, etc.), il y a un besoin inhérent de les combiner. Il y a deux approches distinctes pour intégrer chaque modalités:

- **La fusion multimodale** - cette approche combine les représentations de chaque modalité dans une nouvelle représentation qui contient l'information unifié des différentes modalités (mais pas nécessairement disjointes) [66, 15, 73]. Les exemples typiques de

telles approches incluent la recherche multimodale de photos personnelles en utilisant à la fois la représentation visuelle (de la photo elle-même) et la représentation textuelle de ses annotations [66], l'utilisation ds représentations visuelles et de la parole en recherche de vidéos [73] et bien d'autres où plusieurs modalités sont disponibles.

- **La transmodalité** - d'autre part la traduction transmodale permet de passer d'une modalité à une autre sans combiner les informations des deux modalités concernées [30, 117]. De telles approches permettent de synthétiser une modalité à partir d'une autre, au niveau de la représentation [30, 116] ou même dans le domaine d'origine [71, 90, 133, 117]. La plupart des approches transmodales offrent également le moyen de faire de la fusion multimodale.

Les deux approches ne sont pas nécessairement décorrélées. La transmodalité ou la fusion multimodale sont très reliées et la fusion multimodale peut grandement bénéficier des méthodes qui se focalisent sur la transmodalité. Transmodalité et fusion multimodale ne sont pas limitées aux modalités présentes au même niveau (e.g., images dans un espace de représentation ou texte dans un espace de représentation; voir figure 5.1) et peuvent être effectuées avec différentes modalités à différent niveaux. Bien qu'il y a de nombreuses combinaisons possibles, la plus intéressante utilise les deux modalités plongées dans un espace de représentation. Nous n'allons pas seulement utiliser des espaces de représentation continus mais nous allons aussi explorer la possibilité d'utiliser directement l'espace original d'une modalité car il est interprétable humainement et cela peut permettre de fournir un aperçu du modèle appris.

0.3 Objectif et vue d'ensemble de la thèse

Nous suivons une progression naturelle, en démarrant avec des réseaux simples capables d'utiliser une seule modalité (texte ou image) and nous progressons vers des architectures plus complexes qui combinent différents type de réseaux de neurones ou différentes entrées pour traiter différentes modalités.

Nous focalisons en premier lieu sur la compréhension du langage oral et plus spécifiquement sur la tâche de détection d'attribut (slot filling) dans les systèmes de dialogue homme-machine. L'algorithme d'apprentissage état de l'art récent pour ce problème était les champs conditionnels aléatoires (CRF). Les récentes avancées en réseaux de neurones ont amené à leur utilisation pour résoudre la tâche de détection d'attribut sur des jeux de données relativement simples [68, 129]. Ce travail est découpé en deux parties. Dans la première, nous évaluons les champs conditionnels aléatoires ainsi que les réseaux de neurones. Nous tentons de vérifier si le gain des réseaux de neurones publié dans la littérature récente provient de la nature même des réseaux à modéliser les séquences ou de la qualité des représentations continues qui encode les observations à l'entrée du réseau. Nous montrons que le gain obtenu dans ce contexte est essentiellement obtenu par la représentation en entrée et non le réseau lui même et que la modélisation des dépendances entre attributs, mieux assurée par les CRF, est crucial pour cette tâche [**interseech2015**]. Nous proposons alors une architecture modifiée de réseau récurrent pre-existant qui modélise mieux ces dépendances [**interseech2017**] et qui définit un nouvel état de l'art sur ces données. Nous montrons notamment que les résultats publiés dans la littérature obtenus sur le corpus ATIS [22] ne sont

pas suffisamment significatifs pour en tirer de fortes conclusion, ceci du à la nature facile de ce benchmark. Nos résultats évalués à la fois sur les corpus ATIS et MEDIA [12], un benchmark proposant un challenge plus relevé, permettent de le confirmer. Dans la seconde partie, nous évaluons différent réseaux récurrents à modéliser les séquences. Nous montrons que les meilleures performances sont atteintes avec des réseaux bidirectionnels et une architecture de type GRU (gate recurrent unit). Nous avons également étudié l'apport d'informations simples non locales aux réseaux avec de l'information provenant du dialogue passé. L'incorporation de d'informations binaires indiquant si un attribut a déjà été mentionné dans le dialogue permet d'améliorer les performances.

Après l'évaluation de méthodes d'apprentissage profond dans un contexte de traitement des langues (et donc de textes), nous nous intéressons à un autre problème monomodal, cette fois-ci en vision par ordinateur: la prédiction du mouvement à partir d'images statiques. Nous évaluons l'utilisation de simples architectures de réseaux convolutifs et déconvolutifs pour générer des prédictions de mouvement sur le corpus d'action humaine ETH [96]. Nous proposons une architecture neuronale simple de type encodeur-décodeur avec une branche supplémentaire qui modélise la différence temporelle entre l'image observée et l'image à prédire. Cette architecture est comparée à un simple encodeur-décodeur convolutif qui génère les anticipations futures à un intervalle temporel fixe et utilisé itérativement pour générer des prédictions à des intervalles de temps supérieur. Nous montrons que notre proposition avec cette branche additionnelle permettant de modéliser le temps peut créer avec succès des représentations qui encode la position et l'orientation d'un personnage humain et que ces représentations peuvent être utilisées pour synthétiser des anticipations réalistes pour un intervalle de temps arbitraire. Nous montrons également que générer des prédictions directement fonctionne mieux que d'utiliser une méthode itérative, à la fois en terme de visualisation humaine et en erreur quadratique moyenne entre l'image prédite et la vérité terrain.

Après les approches monomodales, nous continuons avec les méthodes d'apprentissage profond qui exploitent plusieurs modalités. Nous commençons par décrire les méthodes de fusion multimodales état de l'art. Nous introduisons ensuite notre proposition de réseau neuronal profond bidirectionnel transmodal (BiDNN) qui permet de faire de la fusion multimodale. En renforçant la symétrie de deux systèmes opposés de traduction transmodale, nous obtenons la création d'un nouvel espace de représentation où les représentations monomodales disjointes de départ peuvent être directement comparées. Nous utilisons cet espace de représentation pour effectuer de la fusion multimodale. Dans la dernière partie, après l'introduction des réseaux conditionnels génératifs adverses (GAN) (texte vers images), nous proposons une méthode pour les utiliser pour faire de la fusion multimodale tout en ayant la capacité de visualiser les traductions multimodales dans l'espace visuel d'origine (l'espace des images).

Après l'introduction des notions théoriques, nous évaluons différentes représentations monomodales pour la modalité parole et visuelle et nous évaluons différentes façons de faire de la fusion multimodales. Nous comparons des autoencodeurs multimodaux à notre réseau BiDNN et nous montrons que notre BiDNN est le nouveau modèle état de l'art sur la tâche de d'hyperliens vidéo (video-hyperlinking). Nous proposons également une méthode utilisant les CGAN pour faire de la fusion multimodale et nous montrons que cette méthode peut produire des représentations de la même qualité, voire meilleure que les BiDNN avec en plus la faculté de produire des visualisations transmodales qui donnent un aperçu interprétable

humainement du modèle appris. Ce modèle est toutefois actuellement limité à la taille des images observées et ne peut rivaliser avec le BiDNN pour la fusion multimodale d'images de taille arbitraire.

Nous discutons également de la tâche de video-hyperlinking en terme de diversité des résultats proposés, à la fois à travers un questionnaire et de mesures objectives. Nous montrons que notre BiDNN, en plus de définir le nouvel état de l'art en terme de précision, permet d'améliorer la diversité des résultats proposés parmi les cibles pertinentes selon les mesures objectives et les évaluations manuelles qui sont corrélées et peut donc être utilisé pour estimer la diversité quand une évaluation humaine n'est pas possible.

Dans ce travail, nous donnons une vue d'ensemble de différentes représentations monomodales qui sont évalues dans les tâches de détection d'attributs, prédiction du mouvement et hyperliens de vidéos. Nous introduisons alors nos principales contributions, un réseau neuronal profond bidirectionnel (BiDNN) utilisé au départ pour la fusion multimodale et défini le nouvel état de l'art en video-hyperlinking. Nous évaluons ce modèle en terme de diversité où nous montrons qu'il offre une diversité amélioré de résultats parmi les ensembles de vidéos pertinentes en plus des performances état de l'art en terme de précision. Dernièrement nous étudions des moyens de visualiser des traductions transmodales tout en faisant de la fusion multimodale et nous proposons d'utiliser les CGAN. Nous utilisons la partie générative du réseau pour synthétiser les visualisations d'un modèle transmodal et la partie discriminante du réseau pour faire de la fusion multimodale.

0.4 Principales contributions

Dans cette dissertation, nous évaluons la pertinence des représentations neuronales pour faire de la fusion multimodale. L'objectif de ce travail intitulé "Architectures neuronales profondes pour l'apprentissage automatique de représentations de données multimédias multimodales" a été d'évaluer l'existant et de proposer de nouvelles méthodes pour l'apprentissage automatique non-supervisée de représentations de données multimodales dans le contexte multimédia. Toutefois, nous avons commencé par étudier des problèmes impliquant une seule modalité.

Le premier objectif a été d'évaluer des architectures pour des entrées monomodales, texte ou visuelle. Pour les entrées textes, le but a été de comparer les méthodes de plongement de mot ou de texte et les architectures pour modéliser les séquences. Pour les entrées visuelles, nous avons fixé le but de prédire le mouvement étant donné une image statique d'une personne faisant une action simple. Le but a été d'évaluer les méthodes et architectures existantes, les combiner, pour fournir une amélioration pour chaque tâche et évaluer leur faisabilité pour les utiliser aux différentes sous-tâches dans le cadre d'architectures multimodales plus complexes présentées dans la suite.

Le premier et second objectif a été de développer et évaluer des architectures neuronales plus complexes qui peuvent traiter différentes modalités, principalement visuelle et textuelle, faire de la fusion multimodale aussi bien que de la transmodalité. Le but a été d'améliorer la recherche multimodale en développant des architectures qui permettent d'obtenir de meilleures représentations multimodales quand on fusionne deux modalité initialement disjointes. La tâche principale sur laquelle nous nous sommes évalué a été la tâche de video hyperlinking, une variation de la recherche multimodale où l'objectif est de retrou-

ver une ensemble de segments vidéos qui peut être intéressant pour la personne visualisant un segment spécifique de vidéo. Comme cette tâche est une tâche présente dans la campagne d'évaluation TRECVID, nous avons aussi participé et évalué nos méthodes lors de cette campagne. Lors de nos recherches pour améliorer la fusion multimodale, nous avons aussi explorer la possibilité de visualiser les modèles appris dans une forme naturelle pour un humain.

En ce qui concerne le premier objectif d'évaluer des architectures monomodales pour modéliser des entrées textuelles ou visuelles, nous avons principalement utilisé deux tâches. Pour le modalité textuelle, nous avons évalué différentes architectures neuronales pour la tâche de détection d'attribut en compréhension de la parole dans le cadre de dialogue téléphonique homme-machine. Nous avons évalué les performances de différentes architectures neuronales, depuis de simples réseaux récurrent, les versions Jordan et Elman au architectures récurrentes plus récentes telles que les LSTM/GRU qui apprends les informations à mémoriser au sein d'une séquence. Nous avons montré que dans ce cadre, les architectures GRU bidirectionnelles étaient les plus performantes bien que les gains des architectures neuronales sur cette tâche étaient principalement dus à la représentation initiale des données en entrée du réseau, plutôt qu'au réseau lui même. Même si ces réseaux récurrent sont adaptés pour modéliser des séquences et mémoriser des informations non locales, elles ne sont pas efficaces à modéliser les dépendances entre les étiquettes de sorties comme par exemple les champs conditionnels aléatoires (CRF) sur cette tâche.

En ce qui concerne la modalité visuelle, nous nous sommes concentrés sur la prédiction du mouvement et nous avons developpé une architecture qui prédit un futur mouvement pour une différence de temps arbitraire depuis une image unique. Nous avons étendue la possibilité des réseaux convolutifs afin de leur donner la capacité d'apprendre une représentation d'une personne qui encode implicitement sa direction et sa posture. basé sur cette représentation, un réseau déconvolutif est capable de synthétiser une prédiction correcte d'un mouvement d'un personne, en anticipant correctement la direction du mouvement et le changement de posture pour un arbitraire, non-discret déplacement temporel.

Pour le second objectif qui est d'améliorer la fusion multimodale, nous avons tout d'abord développé une nouvelle architecture (un réseau profond bidirectionnel) qui, contrairement aux autoencodeurs multimodaux existants, se concentrent sur la traduction transmodale et crée un espace de représentation commun pour les deux (texte vers image et image vers texte) traductions transmodales. Ce nouvel espace de représentation est alors utilisé pou faire de la fusion multimodale. Nous avons montré sur plusieurs évaluations et différentes monomodales représentations en entrée que notre méthode fournit une représentation multimodale qui améliore significativement la représentation obtenue avec des autoencodeurs multimodaux. En plus de ces évaluations, nous avons participé à l'évaluation de la campagne internationale TRECVID où notre méthode est arrivé en tête définissant un nouvel état de l'art. Nous avons aussi analysé notre méthode en terme de diversité de résultat à travers une évaluation manuelle (questionnaire web) et montré que notre méthode offre de bonnes performances à la fois en terme de précision et de diversité.

Dans la dernière partie, nous avons évalué la possibilité d'utiliser des réseaux conditionnels génératifs adverses (CGAN) pour faire de la fusion multimodale tout en préservant la capacité de synthétiser dans le domaine original (le domaine des images) dans le but d'offrir des visualisations des traductions transmodales apprises. Nous avons montré que les CGAN peuvent être utilisés pour la fusion multimodale et sont à l'état de l'art pour des

images de petite taille. Cependant ils sont très compliqués à entraîner et limités en terme de taille d'image qu'ils peuvent traiter. Pour cette raison, ils ne peuvent actuellement concourir contre des autoencodeurs multimodaux ou notre BiDNN mais peuvent offrir de belles visualisations du modèle transmodal.

0.5 Explorations futures

Plusieurs directions peuvent être explorées. En ce qui concerne la détection d'attributs en compréhension de la parole, des architectures plus complexes qui modélisent partiellement ou complètement les dépendances des sorties ont à être étudié. De telles architectures peuvent varier de simples connections récurrentes aux décisions prises dans le passé aux architectures qui prédisent plusieurs sorties à la fois, avec une fonction de coût modifiée qui modélisent les probabilités de la séquence de sortie.

En fusion multimodale, spécialement pour la tâche de video-hyperlinking, il y a aussi de mutiples chemins à explorer. Notre réseau BiDNN est le nouvel état de l'art mais est toujours relativement simple. Explorer de possible améliorations ou le combiner avec d'autres méthodes peut être payant. l'améliorer peut passer par une fonction de côut additionnelle qui force les deux couches centrales à être aussi similaire que possible ou en ajoutant des couches pour permettre l'apprentissage de bout en bout.

Nous espérons qu'un fonction de coût additionnelle améliore la symétrie des projections sans trop rigidifier et étrangler le réseau, le fait d'introduire plus de variables partagées devrait le permettre. l'apprentissage de bout en bout est en général plus performant et il peut être appris de manière plus fine à tous les niveaux et donc nous pouvons ici aussi espérer au moins de petites améliorations.

D'autres façons plus avancées d'améliorer le BiDNN peuvent être également l'utilisation d'autoencodeurs variationnels [123], qui ont de meilleures propriétés statistiques de modélisation, et les connecter à l'entrée de l'architecture faite pour améliorer la recherche monomodale. Les réseaux génératifs adverses sont une nouvelle et prometteuse piste de recherche. Tandis que nous avons montré leurs potentiels pour la tâche de video-hyperlinking, ils sont actuellement grandement améliorés. Comme les architectures qui sont capables de gérer de grandes images, il serait intéressant d'évaluer s'ils vont être capables de passer à l'échelle comme le BiDNN.

Nous avons également seulement évalué une traduction transmodale qui va des représentations de parole au domaine des images. Il serait intéressant d'évaluer un modèle de bout en bout qui va directement du domaine de la parole au domaine de l'image et vice versa. Ceci pourrait être possible en ajoutant une couche récurrente d'un côté à la place de la représentation de la parole, en laissant tel quel le reste du générateur. La direction opposée devrait être aussi facilement modélisée en utilisant un réseau convolutif, avec du bruit aléatoire en entrée, et un RNN pour générer des phrases réalistes à partir d'un visuel donné. La performance d'une telle architecture reste à explorer.

Chapter 1

Introduction

Contents

1.1 Context

With the recent resurgence of neural networks, the rapid development of deep learning methods and the proliferation of massive amounts of unlabeled data, unsupervised learning algorithms — which can automatically discover interesting and useful patterns in such data — have gained popularity among researchers and practitioners [14]. Modern deep learning methods and recent advances in hardware allow to bypass careful, and often time consuming, manual feature engineering and allow for end-to-end neural methods, that learn the primary task, as well as the representations needed by it, to be easily deployed. A good indicator of the popularity and success of deep learning methods is the ImageNet challenge where, since 2013, almost all the participants used deep learning methods [92] and the few participants using handcrafted representations were lagging behind. Deep learning, and thus automatic representation learning, is not only an increasingly popular trend in computer vision. Other fields, such as natural language processing, speech recognition, multimodal retrieval systems and many others follow this trend.

The goal of research has moved from designing features and combining classifiers to developing neural architectures that learn meaningful representations and are able to fully exploit them for a multitude of different tasks. Many different kinds of deep learning architectures exist and each is specifically tailored for tackling a set of problems. In natural language processing and sequence modeling, recurrent neural networks were made after the necessity of having neural networks that can model sequences of varying lengths. The

latest models of such networks also include gates that allow them to learn which information to retain and which to forget, and define the state of the art in spoken language understanding [129], neural translation [4], question answering [125] and many other tasks that rely on sequence modeling and selective information retention. In computer vision, convolutional neural networks are used for implicitly obtaining good translation and scale invariant features [101] and are used for many computer vision tasks such as classification [61], semantic segmentation [64], saliency estimation [41] and many others. When transposed, they are often called "deconvolutional" networks and are used for generating synthetic images from an existing representation or from a source of random noise.

The biggest advantages come when combining multiple different types of neural networks into bigger, more complex architectures that are able to incorporate different input sources, learn meaningful representations, merge them or transform them and excel at different tasks. This is most interesting in unsupervised learning where such networks can be trained by exploiting the big magnitude of unlabeled data available on the Internet, without having the need for human annotators to perform the tedious work of providing labels for a deep learning network to learn form. Typical examples include unsupervised multimodal processing of unlabeled videos where deep learning networks are able to fuse information obtained from automatic speech transcripts and visual representations from convolutional neural networks in order to assess the similarity of different videos. This specific task has been of growing interest in yearly international benchmarking initiatives like MediaEval and TRECVID. Other typical learning tasks where different types neural architectures include, but are not limited to, motion prediction [121, 123], generating images from text [89, 133], image inpainting [131, 134] and many others.

In this dissertation, we evaluate the thesis that neural embeddings are well suited for multimodal fusion. We mainly focus firstly on developing neural architectures that exploit either solely textual or visual information and secondly on architectures that exploit the combination of textual and visual information. We evaluate different recurrent architectures for spoken language understanding and compare the state of the art; we develop simple convolutional architectures that predict motion from static images, thus filling the gap between classical machine learning methods and more heavy deep learning methods. Regarding unsupervised methods to fuse textual and visual information, we elaborate and develop different architectures that are able to better exploit big video collections and improve retrieval, both in terms of accuracy and diversity of the retrieved collection. All the proposed methods are evaluated with benchmarks such as TRECVID and define the new state of the art. Additionally, we also evaluate methods to better visualize learned deep learning models in a human understandable form.

1.2 Organization of the Manuscript and Contributions

This manuscript follows a natural progression starting from simple deep learning methods able to utilize only one modality (text or images) and slowly progressing towards more complex architectures that combine different types of neural networks or different inputs to complex and heavy networks that consist of different neural architectures combined in a dynamic way that utilize and synthesize different modalities.

In Chapter 3, we focus on spoken language understanding and, more specifically, on the

problem of slot filling. The state of the art method for the slot filling task were conditional random fields. However, recent advances in neural networks lead to the use of recurrent neural networks for slot filling on simple datasets [68, 129]. This chapter consists of two parts. In the first part we thoroughly evaluate conditional random fields and recurrent neural networks. We start by assessing whether recurrent neural networks do gain from their use of continuous representations as inputs or from their capability to model sequences. We show that, while continuous representations bring initial improvement, the ability of conditional random fields to model output-label dependencies is crucial for the task of slot filling. We then propose a new, modified, architecture that models output label dependencies with a learned representation space and defines the state of the art today. Additionally, we show that results obtained on the classic ATIS dataset [22] are not significant enough to draw conclusions and that results obtained on a more challenging dataset, like MEDIA [12], are required. In the second part, we evaluate different gated recurrent neural networks, possible directions of modeling sequences, and adding contextual information to a recurrent architecture. We show that the best performance is achieved with bidirectional sequence modeling, with architectures using gated recurrent units (GRUs) and by incorporating additional binary contextual information that indicates whether a specific concept was already mentioned within the current dialog or not.

In Chapter 4, we focus on another single-modal problem, this time in computer vision: action forecasting from static images. We evaluate the feasibility of using simple convolutional and deconvolutional architectures for generating movement predictions on the ETH human actions dataset [96]. We propose a simple encoder-decoder neural architecture with an added branch that models the temporal difference between the image provided as input and desired prediction. This architecture is then compared to a simple convolutional encoder-decoder architecture that generates future anticipations at a fixed time interval and is used iteratively to generate predictions at larger time intervals. We show that a simple convolutional neural architecture with an added fully-connected branch for time modeling can successfully create representations that encode the stance and orientation of a human character and that those representations can be then used to synthesize realistic anticipations for the given arbitrary temporal interval. We additionally show that generating predictions directly, in one step, performs better than using an iterative method, both in terms of manual visual evaluations and mean square error between the predicted image and the groundtruth.

Chapter 5 gives a theoretical overview of standard deep learning architectures that utilize two modalities and either translate from one modality to another or combine them by performing multimodal fusion. We start by describing the existing state-of-the-art methods for multimodal fusion. We then introduce our proposed bidirectional deep neural networks where we focus on crossmodal translations as means of performing multimodal fusion. By enforcing partial symmetry of two crossmodal translations, we create a new representation space where both, initially disjoint, modalities can be directly compared. We then use this representation space to perform multimodal fusion. In the last part, after introducing conditional (text to image) generative adversarial networks, we propose a method to use them for performing multimodal fusion while also having the ability to visualize crossmodal translations in the original visual domain - the image space.

In Chapter 6, we evaluate our propositions from Chapter 5 within the task of video hyperlinking. We evaluate different single-modal representations for the speech and visual modalities and we evaluate different ways of performing multimodal fusion. We compare

multimodal autoencoders to our proposed bidirectional deep neural networks and we show that bidirectional deep neural networks (BiDNN) define the new state of the art in video hyperlinking both in offline and live evaluations. Furthermore, we propose a method of using conditional generative adversarial networks for performing multimodal fusion where we show that they can produce embeddings of the same quality as bidirectional deep neural networks or better in addition to producing crossmodal visualizations that give human-interpretable insights into the trained video-hyperlinking model. Conditional generative adversarial networks are currently limited in regards to image size and cannot currently outperform bidirectional deep neural networks for multimodal fusion given images of arbitrary size. In Chapter 7, we also discuss video hyperlinking in terms of provided diversity, both through an online questionnaire and through intrinsic measures. We show that our proposed bidirectional deep neural networks, in addition to defining the new state of the art in terms of precision, enable increased diversity within the proposed set of relevant targets and that intrinsic evaluations of diversity correlate with manual human evaluations and can be used for estimating diversity when a human evaluation framework is not available.

In this work we give an overview of different single modal representations that are evaluated in the tasks of slot filling, action forecasting and video hyperlinking. We then introduce our main contribution, bidirectional deep neural networks (BiDNN) that are used primarily for multimodal fusion and define the new state of the art in video hyperlinking. In addition to evaluating relatedness, we also evaluate bidirectional deep neural networks in terms of diversity where we show that they also offer improved diversity of the recommended set of videos in addition to the state-of-the-art performance in terms of precision. Lastly, we investigate ways of visualizing learned crossmodal translations while also performing multimodal fusion and we propose to use conditional generative adversarial networks. We use the generative part of the network to synthesize visualizations of a crossmodal model and the discriminative part to perform multimodal fusion. Possible future work could include, but is in no way limited to incorporating additional losses and restrictions to further improve bidirectional neural networks and extending both bidirectional neural networks and conditional generative adversarial networks to perform end-to-end learning. Additionally, in the case of conditional generative adversarial networks, it would be necessary to evaluate more complex and recent architectures that can use and synthesize bigger images and evaluate whether they can still perform comparably to bidirectional deep neural networks or not.

Chapter 2

Continuous Representation Spaces

Contents

Recent advances in deep learning have changed the way we tackle problems in natural language processing, computer vision and multimedia. Manual feature engineering has lost its importance to automatic representation learning methods. In text, bag-of-words methods have been replaced with methods that learn representations in an autonomous, unsupervised way from available text corpuses [70, 59]. In computer vision the same has occurred and manually designed features [65, 6, 69] are now replaced with self-learned representations that are readily available with deep learning methods at no additional cost [98]. In multimedia, the same trend followed where methods for fusing different textual, visual and other modalities have been developed [55, 73, 30, 15].

In order to use various input modalities (text, images, etc.) with deep learning methods, there is a need to embed each input into a continuous representation space. The process of embedding an input modality, not only makes it more suitable for neural networks to handle but typically also provides useful intrinsic properties [70, 98, 85, 67]. In this chapter, we introduce the basic neural architectures we will extensively use in this work and allow representation learning of text and images, translations between different representations as well as synthetization of artificial samples into the original domain (e.g., image space or text) given a continuous representation.

2.1 Representing Textual Information

When dealing with text, we typically focus on chunks of different lengths: single words or textual segments of varying length that can be part of sentences, multiple sentences, paragraphs or entire documents. Also, we can treat multiple words either as independent wholes (by representing the entire chunk) or as sequences of words. In this section, we illustrate methods that enable each of the different approaches to modeling textual information.

2.1.1 Representing Words

Although different methods exist for generating continuous word representations [78, 103], we focus solely on *Word2Vec*, as it provides the best accuracy on a multitude of tasks and datasets [70]. *Word2Vec* is a group of models, based on a shallow, two-layer feed-forward neural architecture, that serves the purpose of generating word embeddings.

The first *Word2Vec* architecture, named continuous bag of words (CBOW) is illustrated in Figure 2.1. Given a window, the idea is to predict the central word from the words preceding it and the words that follow it. Each word is represented with a vector $\mathbf{w}$ stored in the matrix $\mathbf{W}$ of size $vocabulary\_size \times representation\_size$. All the words within a window, except for the middle one that is ought to be predicted, are then summed to a common fully-connected layer named the projection layer. From there, the output layer tries to reconstruct the vector representing the middle word. The fact that all the input word representations are summed makes their order irrelevant and names the architecture continuous bag of words, since this is an analogue property in discrete bag of words. To illustrate on an example, with the phrase "the cat is on the couch", and a window of size 5 (2 words before and two words after), the first window is "the", "cat", "is", "on", "the". The words "the", "cat", "on", "the" are presented as inputs to a *Word2Vec* DBOW architecture, where they are embedded to their $\mathbf{w}_2$, $\mathbf{w}_3$, $\mathbf{w}_1$, $\mathbf{w}_2$, respectively. The word "is" is also embedded and its vector $\mathbf{w}_4$ is expected as output. Learning is performed through backpropagation, starting from the difference of the expected output. The next window would then be "cat", "is", "on", "the", "couch" with the word "on" being predicted, given the other words from the window. All the word representations $\mathbf{w}$ are initially randomly initialized (thus the whole matrix $\mathbf{W}$) and are later updated through backpropagation. This makes words appearing in similar contexts have similar vectors, which is a nice property of *Word2Vec* [70]. Additionally, the authors propose also negative sampling - where words that do not belong together are randomly chosen and their representations are pushed further apart.

The second *Word2Vec* architecture, named skip-gram, is illustrated in Figure 2.2 and is similar to CBOW (a 2-layer, feed-forward neural architecture) but inverted: with the skip-gram model, the idea is to predict words of the near context given a specific word. In the previous example, with the current window "the", "cat", "is", "on", "the", given the word "is", the words "cat" and "on" are to be predicted, as well as "the". However, the more distant the words are, the less they are related to the current word. The authors suggest a context window of size 5 to 10 words [70].

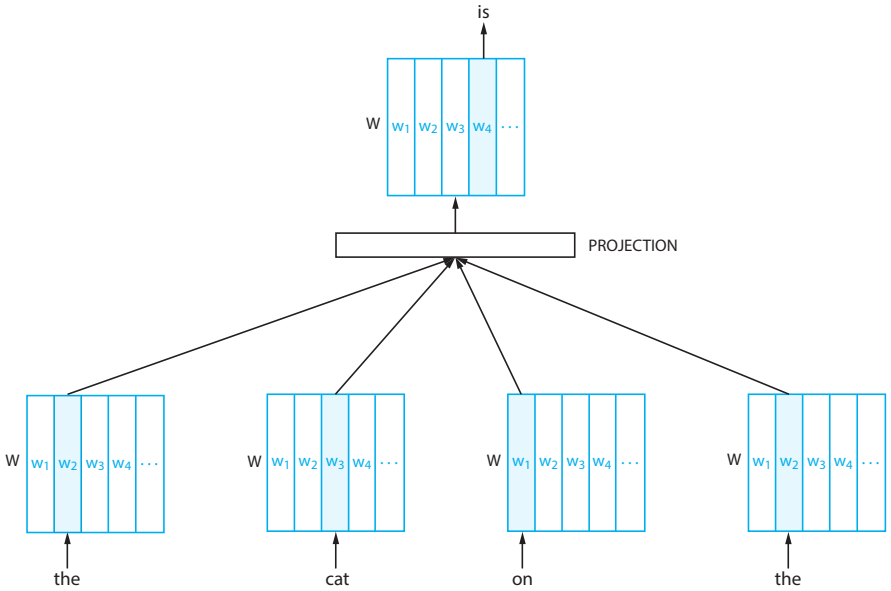


Figure 2.1: Word2Vec - continuous bag of words (CBOW): the central word of a window is predicted given the words on the left and the words on the right. In the illustration, the window "the", "cat", "is", "on", "the" is currently processed.

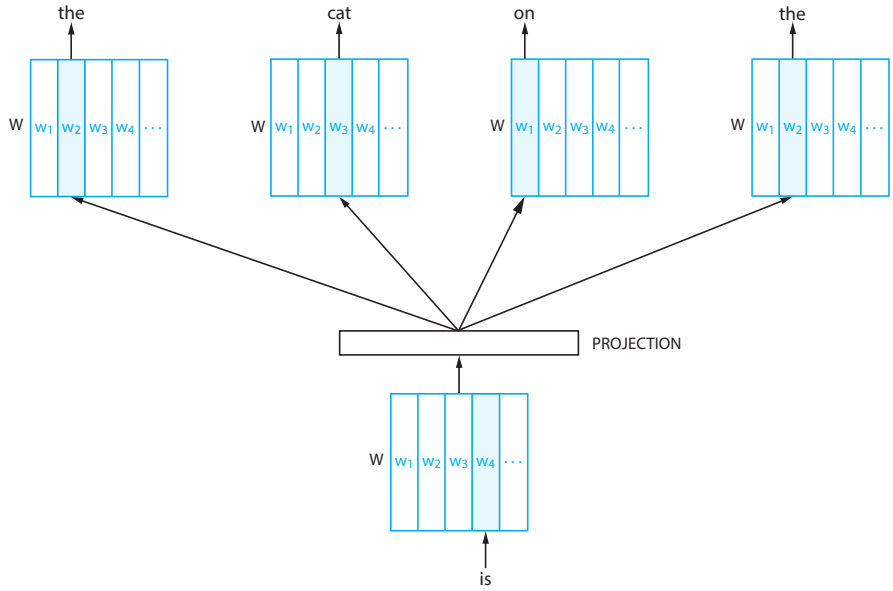


Figure 2.2: Word2Vec - skip-gram: given a word within a window, its context is predicted. In this illustration, the window currently processed is again "the", "cat", "is", "on", "the".

2.1.2 Representing Textual Segments

In many scenarios, it is not necessary to represent single words but a chunk consisting of multiple words. Chunks can vary from segments of a sentence or just a few sentences (e.g., automatic speech transcripts of very short video segments), through paragraphs to entire documents. The simplest way to represent textual segments is to aggregate the representations of its individual words and one of the simplest and most effective to aggregate the representations of each word is to compute the average of their average *Word2Vec* representation [13].

A more complex and dedicated way of representing chunks of text are paragraph vectors [59]. Their architecture is inspired by word representation methods (*Word2Vec*) with which they share a lot of similarities. Paragraph vectors are specifically tailored to represent textual chunks like paragraphs or entire documents and just like *Word2Vec* exist in two variants. The first variant, named paragraph vector distributed memory (PV-DM), is illustrated in Figure 2.3. Each paragraph is mapped to a vector $\mathbf{p}$, stored as a column in matrix $\mathbf{D}$. Each word is also mapped to a vector $\mathbf{w}$, stored as a column of matrix $\mathbf{W}$. Paragraph vectors are not shared over different paragraphs (there are as many columns in matrix $\mathbf{D}$ as there are paragraphs in the dataset) and word vectors are shared across all paragraphs. The system is setup so to predict the next word within a window, given the window itself and the paragraph that precedes it. Given an example where it is necessary to predict the word "on" given the window "the", "cat", "is" and the previous paragraph, first all the vectors of the given words inside the window and of the previous paragraph are extracted from their respective matrices $\mathbf{D}$ and $\mathbf{W}$ (just like a lookup table) and are then concatenated. The vector is then passed through a fully-connected softmax layer that generates a prediction of the next word as a multiclass classification problem. More formally, PV-DM is defined as follows:

$$\mathbf{y} = \mathbf{U}(\mathbf{p}_i \| \mathbf{w}_{t-k} \| \cdots \| \mathbf{w}_{t+k}) + \mathbf{b} \tag{2.1}$$

where $\mathbf{U}$ and b are learnable parameters of the softmax layer and the $\|$ denotes concatenation. The authors also mention the possibility of averaging the representation [59] but do not explore that possibility. The paragraph and word representations in the $\mathbf{D}$ and $\mathbf{W}$ matrices respectively are updated through backpropagation. After training, each vector of $\mathbf{D}$ is used to represent its respective paragraph. The matrix $\mathbf{D}$ is seen as a distributed memory that remembers the necessary context (the embedded preceding paragraph) for the currently analyzed window and gives the method its name. Although the name implies the method is used for representing paragraphs, it is equally possible to represent entire documents in the same manner [59].

The other variation of paragraph vectors is named distributed bag of words (PV-DBOW) and is illustrated in Figure 2.4. This model resembles the *Word2Vec* skip-gram model and is lighter compared to the previous PV-DM model as only one matrix ($\mathbf{D}$) is stored. The model is trained by sampling random words from a window and trying to predict them (as a multiclass classification problem) given the paragraph as input. This method, although lighter, performs less well than PV-DM according to the authors [59] and to our experiments in Chapter 6.

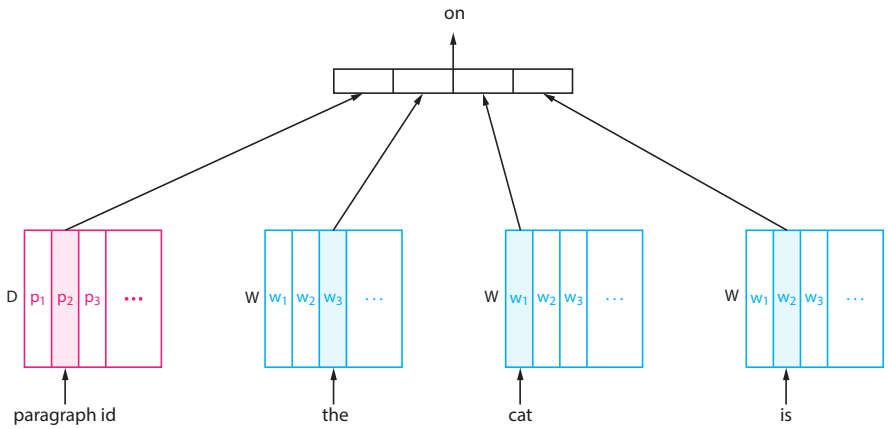


Figure 2.3: PM-DM - the distributed memory version of the paragraph vector model where the next word is predicted given the words within a window and the paragraph preceding them that forms a distributed memory storing the necessary context.

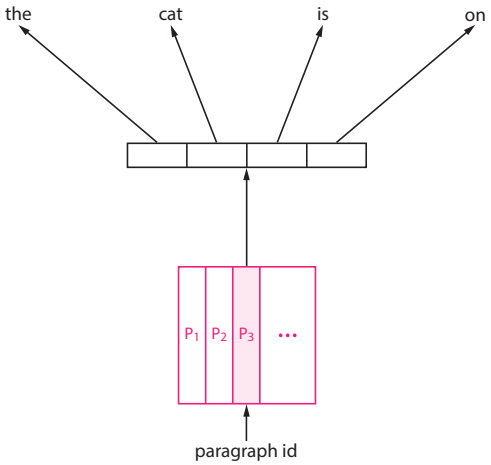


Figure 2.4: PM-DBOW - the distributed bag of words version of the paragraph vector model where words sampled randomly from a window are predicted given a paragraph.

2.1.3 Representing Textual Sequences

While sometimes it is sufficient to either represent words [70] or represent whole chunks containing words (be it parts of sentences, multiple sentences, paragraphs or entire documents) [59], in many tasks it is necessary to model text as a sequence of words. This is the case in spoken language understanding [129], slot filling [115], as we will see in Chapter 3 and machine translation [17] systems.

Sequences, contrary to many other types of information (e.g., images, measurements, descriptors) often come in a form of variable length. Typical neural networks like a multi-layered pereceptron are tailored to data where each dimension of a sample is fixed and do not work well with samples of variable length across one or more dimensions. Recurrent neural networks were specifically designed to deal with sequences of variable lengths. By adding a recurrent connection to the hidden node itself, the network is able to deal with

variable sequences by simply "unrolling" the recurrent connection the required number of times. We discuss two main categories of recurrent neural networks: simple recurrent neural networks and gated recurrent neural networks. The two groups of architectures differ in their capability of modeling long sequences and by their complexity.

2.1.3.1 Simple Recurrent Neural Networks

The simple recurrent network (SRN) was introduced by Elman [27] in an architecture that bears his name: the Elman network. The Elman network is defined as follows:

$$\mathbf{h}_t = act_1(\mathbf{W}_h\mathbf{h}_{t-1} + \mathbf{W}_x\mathbf{x}_t + \mathbf{b}_h) \tag{2.2}$$
$$\mathbf{o}_t = act_2(\mathbf{W}_o\mathbf{h}_t) \tag{2.3}$$

We denote the current hidden state of a recurrent neural network as a vector $\mathbf{h}_t$, the current input as a vector $\mathbf{x}_t$ and the current output as vector $\mathbf{o}_t$. The Elman network, as illustrated in Figure 2.5 defines the current hidden state $\mathbf{h}_t$ as a combination of the previous hidden state $\mathbf{h}_{t-1}$ and the current input $\mathbf{x}_t$, as denoted in Equation 2.2. The matrix $\mathbf{W}_h$ is sometimes denoted as a context unit as it serves the purpose of modeling the influence of the previous hidden states or, in other words, the previous context of the currently analyzed input.

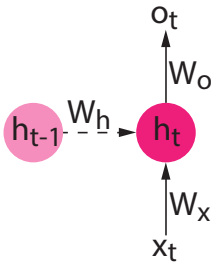


Figure 2.5: An Elman network is a simple recurrent neural network where the current hidden state h_t is defined by the current input x_t and the previous hidden state h_{t-1}.

A variation of the Elman network architecture is the Jordan Network architecture [47]. Both the Elman and the Jordan recurrent neural network architectures are considered simple recurrent networks, however, the Jordan architecture defines its current state by the current input and the previous output (contrary to the Elman architecture that defines it in regards of the previous state), as illustrated in Figure 2.6. The Jordan network architecture is defined as follows:

$$\mathbf{h}_t = act_1(\mathbf{W}_h\mathbf{o}_{t-1} + \mathbf{W}_x\mathbf{x}_t + \mathbf{b}_h) \tag{2.4}$$
$$\mathbf{o}_t = act_2(\mathbf{W}_o\mathbf{h}_t) \tag{2.5}$$

The only difference between Equation 2.2 and Equation 2.4 is the first product, where the context matrix $\mathbf{W}_h$ models the influence of the previous context through the previous output $\mathbf{o}_{t-1}$ and not through the previous hidden state $\mathbf{h}_{t-1}$, as in the Elman network architecture.

Both in Figure 2.5 and in Figure 2.6, we illustrated one unit only. In a typical recurrent neural network a layer of multiple units would be formed, where each unit would behave as described.

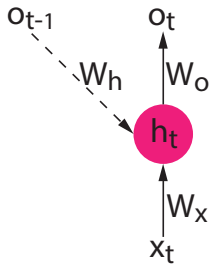


Figure 2.6: A Jordan network is a simple recurrent neural network where the current hidden state h_t is defined by the current input x_t and the previous output o_{t-1}.

2.1.3.2 Gated Recurrent Neural Networks

In practice, simple recurrent neural networks have difficulties modeling long-term dependencies [7]. To tackle this problem, gates were introduced in recurrent neural networks [40], allowing them to learn model dynamic information retention and removal.

Long short-term memory networks [40, 34] introduce a series of gates (input gate, forget gate and output gate) that help model the information retained by the recurrent network. A simple LSTM cell is illustrated in Figure 2.7 and is defined as follows:

$$\mathbf{f}_t = act_1(\mathbf{W}_f[\mathbf{h}_{t-1}\|\mathbf{x}_t] + \mathbf{b}_f) \tag{2.6}$$

$$\mathbf{i}_t = act_1(\mathbf{W}_i[\mathbf{h}_{t-1}\|\mathbf{x}_t] + \mathbf{b}_i) \tag{2.7}$$

$$\widehat{\mathbf{C}}_t = act_2(\mathbf{W}_c[\mathbf{h}_{t-1}\|\mathbf{x}_t] + \mathbf{b}_c) \tag{2.8}$$

The forget gate, denoted as a vector $\mathbf{f}_t$, and the input gate, denoted as a vector $\mathbf{i}_t$, both take the previous hidden state $\mathbf{h}_{t-1}$ and determine how much of the previous cell state $\mathbf{C}_{t-1}$ is attenuated ("forgot") and how much is the state influenced by the new input. In the given equations, $\|$ denotes concatenation and $\widehat{\mathbf{C}}$ represents the new candidate value for the LSTM cell state. The cell state is updated by first being multiplied with the value of the forget gate, thus attenuating it and then by having the new candidate cell state subtracted after it has been modulated by the input gate:

$$\mathbf{C}_t = \mathbf{f}_t C_{t-1} + \mathbf{i}_t \widehat{\mathbf{C}}_t \tag{2.9}$$

Finally, the current output $\mathbf{o}_t$ and hidden state $\mathbf{h}_t$ are updated by the output gate:

$$\mathbf{o}_t = act_1(\mathbf{W}_o[\mathbf{h}_{t-1}\|\mathbf{x}_t] + \mathbf{b}_o)\vec{h}_t = \mathbf{o}_t act_2(\mathbf{C}_t) \tag{2.10}$$

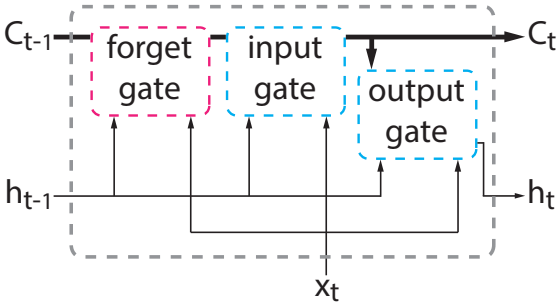


Figure 2.7: A long short-term memory (LSTM) cell consists of a cell state C and a hidden state h, as well as the forget, input and output gates.

Long short-term memory networks are very good at modeling both short and long term dependencies but consist of a relatively complex architecture with many gates and multiple internal states. Gated recurrent units (GRU) [18] are a more novel variation of long short-term memory networks that were created to simplify their architecture while maintaining or improving their long and short term modeling capabilities. They combine the forget and input gates into one update gate and merge the hidden state and cell state into one state, as illustrated in Figure 2.8. More formally, they are defined as follows:

$$\mathbf{z}_t = act_1(\mathbf{W}_z[\mathbf{h}_{t-1} \| \mathbf{x}_t]) \tag{2.11}$$

$$\mathbf{r}_t = act_1(\mathbf{W}_r[\mathbf{h}_{t-1} \| \mathbf{x}_t]) \tag{2.12}$$

$$\widehat{\mathbf{h}}_\mathbf{t} = act_2(\mathbf{W}[\mathbf{h}_{t-1} \| \mathbf{x}_t]) \tag{2.13}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t\widehat{\mathbf{h}}_\mathbf{t} \tag{2.14}$$

where $\mathbf{r}_t$ is a reset gate and $\mathbf{z}_t$ is an update gate. GRUs have been shown to perform better than regular LSTMs while also being faster due to a simpler architecture [20].

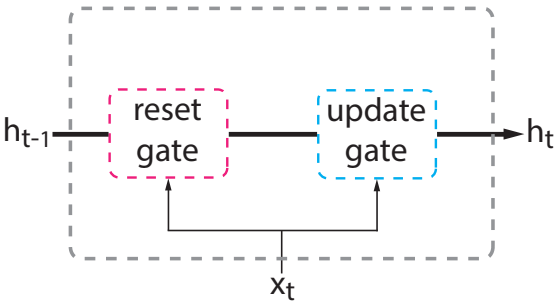


Figure 2.8: In a gated recurrent unit (GRU) cell, the cell state and the hidden sate have been merged into a single hidden state and the input and output gate are replaced with a single update gate.

2.2 Representing Visual Information

To be able to use images with any machine learning algorithm, there is a need to represent either the whole image or different image patches with a robust representation that is suitable for the task. An image can be either described with a set of visual concepts, indicating what is present in the image, or with one or more continuous representations. Continuous representations can be created and used explicitly or they can be learned and used implicitly within an end-to-end neural network.

2.2.1 Convolutional Neural Networks

Methods for obtaining low level and high level representations have become less necessary after the development of deep learning methods since convolutional neural networks. Convolutional neural networks [61] are feed-forward neural networks and variation of a multilayered perceptron that, in addition to fully-connected layers, contain convolutional layers. A single neuron of a fully-connected feed-forward neural network, consisting of three inputs and one output, is illustrated in Figure 2.9 and defined as:

$$o = f(w_1 x_1 + w_2 x_2 + w_3 x_3 + b) \tag{2.15}$$

where w represent weights of each axon, which scale the inputs, and f represents an activation function that maps the output to a restricted codomain and keeps the network within a stable range. b is a bias factor that provides the ability to offset the weighted input and o is the output of the current artificial neuron. For practicality and efficiency (vectorizing the computation), it is possible to incorporate b as an additional weight w (e.g., w_0) and write Equation 2.15 as:

$$o = f(\mathbf{wx}) \tag{2.16}$$

A neural network typically consist of multiple layers and multiple neurons per layer, and not just one, which makes it convenient to define a whole fully-connected layer as:

$$\mathbf{h_j} = f(\mathbf{W_j h_i}) \tag{2.17}$$

where $\mathbf{h}_i$ is a vector representing the output of the previous hidden layer ($\mathbf{x}$, in case it is the first layer), $\mathbf{W}_j$ is a matrix representing all the weights of the current hidden layer j and $\mathbf{h}_j$ is a vector representing the output of the current hidden layer j (or $\mathbf{o}$, in case it is the last, output layer).

A convolutional neural network is still a feed-forward network (its connections propagate forward and there are no recurrent or otherwise cyclical connections) but typically contains a series of convolutional and pooling layers in addition to fully-connected layers. A convolutional layer, followed by a pooling layer is illustrated in Figure 2.10. Convolutional layers are a simple reconfiguration of a fully-connected layer where weights are shared and

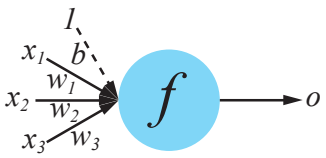


Figure 2.9: A simple neural unit, consisting of 3 inputs x that are weighted by the three weights w of each axon before being summed and passing through an activation function f. An additional weight b (a bias factor) is allows for offsetting the weighted input.

define a convolutional kernel. Convolving an input matrix or, in all generality, a tensor consists of scanning the input with the kernel and summing all the element-wise multiplications each time. This is exactly the same as in a fully-connected layer where we multiply the input vector with the weight vector and then sum all the elements. However, the kernel is used multiple times, which is equivalent to sharing the same weights $\mathbf{w}$ with multiple, different inputs. The convolving kernel forms a visual receptive field that is well tailored for image processing and that offers translation invariance [61].

In the example illustrated in Figure 2.10, the input consists of a 5×5 matrix (e.g., a monochromatic image) and is then convolved with a 2×2 kernel. The kernel starts in the first column of the first row where each element is multiplied with the respective element of the input and the result is then summed. In the given example, $1 \cdot 0 + 1 \cdot 1 + 0 \cdot 0 + 2 \cdot 1 = 1 + 2 = 3$. The kernel then slides to the next column where the next value is computed: $1 \cdot 0 + 1 \cdot 1 + 2 \cdot 0 + 1 \cdot 1 = 1 + 1 = 2$. The process then continues until the whole input is passed and the convolution is finished. The values of the kernel are, just like the weights of a fully-connected layer, learned during training and updated with backpropagation. This allows a neural network to learn meaningful filters automatically [61].

A pooling layer is, on the other side, a layer that contains no learnable parameters (it has no weights or kernels) and it serves the purpose of reducing the dimensionality of the previous layer, thus adding some translation invariance. The most common pooling methods are max-pooling and average pooling. Figure 2.10 illustrates a max-pooling layer that follows a convolutional layer. A max pooling layer simply selects the maximum value of a given region to represent the whole region in the next layer. In this example, the region in the top-left corner is represented with the value of 4 as that is the maximum value of the region. The location of the maximum is remembered so that backpropagation can be continued to the layers preceding a pooling layer. An average pooling layer represents each region with the average value of the region. In this case, during backpropagation the derivatives are distributed to each element of the previous layers after being weighted by their respective sizes.

A typical convolutional neural network consists of a series of interchanging convolutional and pooling layers, followed by a series of fully-connected layers, as illustrated in Figure 2.11. In this work, we solely use the AlexNet [51] architecture and the VGG-16 and VGG-19 architectures [101] that contain 8, 16 and 19 learnable layers (layers that contain weights/kernels that can be updated during learning, namely convolutional and fully-connected layers) respectively though many deeper convolutional neural networks exist [107, 108, 19] that are out of the scope of this work.

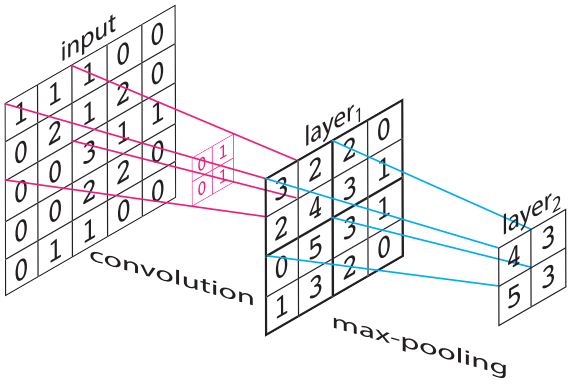


Figure 2.10: A typical convolutional neural network consists of a series of interchanging convolutional and pooling layers. In the illustration: A 2D input, followed by a convolutional layer and then by a max-pooling layer.

2.2.2 Low Level and High Level Image Representations

Although convolutional neural networks were initially developed with the task of supervised classification in mind, they offer representations at different levels of abstraction that are superior to manually designed representations [98]. Related to the supervised classification task of convolutional neural networks, they also provide visual concepts as their output in the last final layer in the form of a binary vector, where each element indicates the presence of one concept and multiple concepts are possible at the same time, defining a multilabel classification task.

Visual concepts are binary indicators that denote the presence or absence of specific objects in an image. The most commonly used set of visual concepts is the one defined by the *ImageNet* project dataset that contains 1,000 object classes in a categorization similar to WordNet [92]. A few examples of ImageNet classes are: *n04557648 - water_bottle*, *n04404412 - television*, *n07749582 - lemon* etc. In addition to standard objects, *ImageNet* provides a fine-grained classification of dogs and cats: *n02124075 - Egyptian_cat*, *n02123394 - Persian_cat*, *n02123597 - Siamese_cat*, *n02085936 - Maltese_dog*, *n02107683 - Bernese_mountain_dog*, *n02094114 - Norfolk_terrier*, etc.

With convolutional neural networks, representations are obtained by presenting an image to the input of a pretrained network and taking its activations in the following manner:

- **low level** are extracted from the initial convolutional layers and aggregated, if necessary, with an aggregation method like fisher vectors [80, 81] or VLAD [44]
- **high level** are extracted from the near-last fully-connected layers and are directly used to represent the whole image
- **visual concepts** are extracted by the last, output layer of the network, given that it was trained on such a classification task (e.g., ImageNet)

Figure 2.11: Three popular convolutional neural network architectures: AlexNet, VGG-16 and VGG-19. Regardless of number of layers the initial convolutional layers provide good low level image embeddings, the fully-connected layers towards the end of the architecture provide good high level embeddings and the last, output layer, provides visual concepts.

2.3 Encoder-Decoder and Autoencoding Networks

In addition to classification tasks and obtaining representations, neural networks are also used to synthesize back from the representation space into the original space (text, image space, etc.) or into a new or the same representation space. Such architectures typically consist of an encoder network and a decoder network, though they are as a whole connected into one network, where learning is performed altogether (backpropagation starts from the decoder and continues into the encoder). A very common example of an encoder-decoder network is a sequence to sequence network [17], used in machine translation and consisting of a recurrent neural network that acts as a decoder and takes a sentence of variable length and encodes it into a fixed size representation, and another recurrent neural network that acts as a decoder and synthesizes a new sentence (potentially of different length) given the fixed size embedded representation from the encoder. Another example is a convolutional encoder-decoder network illustrated in Figure 2.12, consisting of a convolutional network that generates an embedded representation given an input image and a "deconvolutional" (transposed convolutional) network that synthesizes back an image, given the original or altered latent representation obtained with the encoder. A convolutional encoder-decoder network can be used to generate predictions directly into the image domain [120]. It is also possible to combine different encoders and decoder: e.g., convolutional encoder with a recurrent neural network as a decoder in order to create an architecture that generates automatic descriptions of images [127].

A special case of encoder-decoder neural networks are autoencoding networks or autoencoders. An autoencoder is a neural network that reconstructs its own input and is used to learn efficient data representations in an unsupervised manner. A typical autoencoder is implemented as a feed forward neural network consisting of an input layer, an output layer and a number of hidden layers with a decreasing number of units towards the center of the architecture, as illustrated in Figure 2.13. Learning is performed with backpropagation, typically with a mean squared error (MSE) loss between the original unaltered input and the

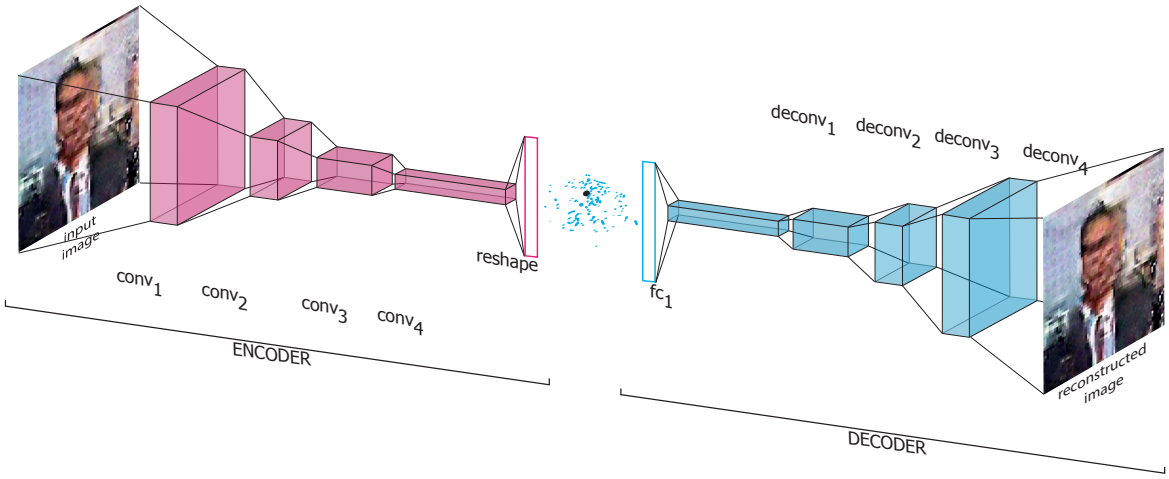


Figure 2.12: A convolutional encoder-decoder architecture. Given an image, a convolutional encoder provides an embedded representation. That latent representation, either modified or not, is then used to synthesize another image with a "deconvolutional" decoder.

generated output. The activations of the central layer are used to obtain a new, compact and efficient representation of the input data. It is important to note that, contrary to methods like principal component analysis, autoencoders converge to a different solution (or a different minima) each time due to their random initialization and the stochastic learning process (e.g., the order of data during training would affect the representations) and thus, two representations from different autoencoders that share the same architecture are not comparable (saving and reusing the weights is however done easily). In addition to learning an efficient and compact data representation, an autoencoder can be used to recover a slightly corrupted or noisy input or represent it with a representation that is more robust to noise. This is done by providing a noisy input to the autoencoder and requiring it to reconstruct the original input. In this setup, the architecture is called a denoising autoencoder.

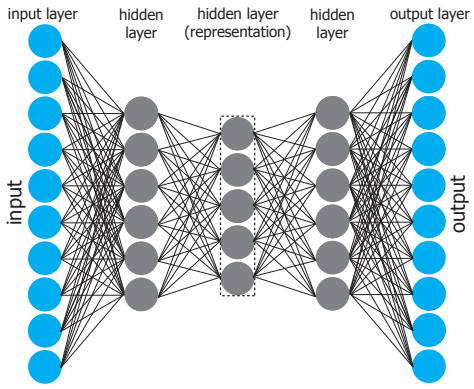


Figure 2.13: A typical autoencoder, consisting of an input layer, three hidden layers and an output layer. The hidden layers contain a smaller number of units thus forcing the network to learn a compressed representation from which the input can be reconstructed.

2.4 Conclusion

The recent resurgence of neural methods changed the way representations are obtained. The focus has shifted from manual feature engineering [23, 65, 6, 69] to developing neural architectures that automatically learn state-of-the-art representations. In this chapter, we gave an overview of the basic neural architectures for representation learning and modification that we will use in this work.

We use word representations as well as sequence modeling methods in Chapter 3 where we will focus on the task of slot filling and carefully analyze which elements are crucial for achieving state-of-the-art results in the task. We will show that good representations and good sequence modeling algorithms are not sufficient if they are not modeling the dependencies of the output labels.

In Chapter 4, we start by the previously illustrated convolutional encoder-decoder architecture that we then extend, in order to generate predictions of human actions from a single image, at arbitrary temporal distances.

Different textual and visual representations, together with autoencoding methods are used in Chapters 5 and 6, where we select the best performing single-modal representations and then proceed to developing new ways of fusing multimodal information that define the new state of the art in video hyperlinking and also allow0 to visualize learned video hyperlinking models in a human interpretable way.

Chapter 3

Spoken Language Understanding - Slot Filling

Contents

In this chapter, we focus on spoken language understanding (SLU) or more specifically, on the slot filling task. The task aims at labeling one or more words into discrete chunks where each chunk can have a separate label depending on its position in a sentence, its adjacent words and related words. For example, in the sentence "*I want a Chinese restaurant near the Tour-Eiffel*", the word "*Chinese*" belongs to a slot labeled as the food-type of a restaurant, and the words "*Tour Eiffel*" as a slot indicating a relative place in Paris.

We first start by introducing the slot filling tagging task and comparing conditional random fields to recurrent neural networks as two possible approaches to address the problem. A careful analysis is performed to locate and emphasize the positive aspects and downsides of each method after which we show that the ability of conditional random fields to model output label dependencies, which is not a capability of recurrent neural networks, is crucial for the task. We then propose a modification of a recurrent neural network architecture that utilizes a learned continuous representation space to improve the modeling of output labels and defines the state of the art today on two datasets.

In the second part, we progress to evaluating both different proposed architectures of gated recurrent neural networks and ways to model sequences. We compare long short-term memory networks (LSTM) and gated recurrent unit (GRU) based networks when performing either forward sequence modeling or bidirectional sequence modeling. After determin-

ing the best performing architecture for the task of slot filling, we improve its performance by incorporating context information.

Typically, in part of speech tagging and slot filling, the IOB (inside, outside, beginning) notation [2] is used to indicate the labels for each word and is used as follows:

- ***B-***<*tag*> is used to mark that the chunk labeled <*tag*> starts from the current word labeled with the *B-tag* label
- ***I-***<*tag*> is used to denote that the current word is part of the chunk labeled <*tag*> that started with one of the previous words of the sequence
- ***O*** is used mark that the current world is **outside** of any chunk; if the previous word was labeled with either a *B-*<*tag*> or an *I-*<*tag*> this label clearly denotes that the previous chunk has ended and the current word does not belong to either the previous or the following chunk

The choice of the tags is deliberate and it varies a lot from dataset to dataset. A typical example of slot filling is as follows: given the sequence of words *{show, flights, from, Boston, to, New, York, today}* there is a corresponding sequence of labels for each word *{O, O, O, B-departing_location, O, B-arriving_location, I-arriving_location, B-date}*, meaning that the first three words and the fifth one are irrelevant for the task (thus marked O), that "Boston" is the departing location, the two words "New York" define the arriving location and "today" indicates the date of the desired flight.

We perform every experiment mentioned in this work on two datasets, namely ATIS [22] and MEDIA [12]. ATIS is a simple, publicly available corpus, used since the early nineties, containing air traffic related phrases. It contains 1,117 unique words and 85 labels in a training set of 4,978 sentences and a testing set of 893 sentences. MEDIA is a more recent and complex, French dataset containing tourist information dialogues. It contains 2,395 unique words 135 labels and 12,908, 1,259 and 3,005 sentences in the training, evaluating and testing sets respectively. Additional information about the datasets can be found in Appendix B.

3.1 The RNN - CRF Dichotomy

Many sequence labeling methods have been investigated in spoken language understanding: SVM [52], HVS [39], machine translation models, finite state transducers and particularly conditional random fields, which have been shown in [38] to be best-suited for this task. Recently, neural networks have been investigated in [68, 129] where they show, on the popular ATIS database, that recurrent neural networks provide state-of-the-art results. Nevertheless, a wide variety of methods are able to provide very good results on ATIS [88], including methods that are not dedicated to sequence labeling (e.g., SVM). These last methods fail [37, 38] when evaluated on MEDIA [12], another SLU database. We thus consider ATIS not to be a very challenging dataset and that the conclusions obtained on this database are not particularly strong and are seldom statistically significant.

Neural networks typically use continuous representations as inputs where the initially symbolic text representations are mapped to a continuous representation space using popular word embedding methods [70, 126]. Such representations has several advantages, the

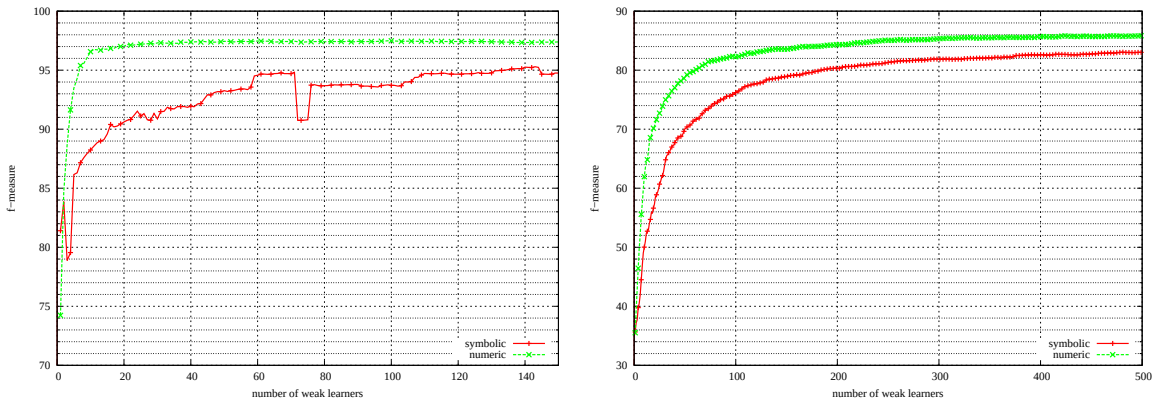
most salient one being the property that words that are syntactically or semantically related are close to each other in the representation space. One question that arises is to know whether improvements come from the representation, the classifier itself or potentially both. However, for slot filling, a precise word clustering is already available: the attribute database linked to the task (e.g., city names, airline names for ATIS, etc.), considering all words from the same class as equal and taking advantage of continuous representations not clear. We thus propose to compare symbolic and embedded inputs under the same classification algorithm, in order to make a strict comparison.

3.1.1 Symbolic Inputs vs Embedded Inputs

For slot filling, in spoken language understanding, input features commonly consist of word observations associated with their relative position from the decision point (the word currently being labeled) in the sequence. For symbolic representations, the feature set is then a bag of pairs "word/relative position" within a specific sliding window of observation. For continuous representations, the feature set is obtained by word embedding methods [70, 126]. The final vector is a concatenation of the embedded representations of each word that belongs to the current sliding window. A common window of $[-2, +2]$ [88, 38] or $[-3, +3]$ [68, 128] is generally sufficient to obtain satisfactory performances. In this work we opted to use a $[-3, +3]$ window for performing the comparisons, although different sizes were tested.

As mentioned earlier, in human-machine applications, database attributes are available to construct a fine clustering of many words supporting concepts: the list of airline names or city names in ATIS or the list of food types, the list of facilities for a hotel and the list of French cities in MEDIA. If a word belongs to such a cluster then a tag representing the cluster is used instead of the word, both when dealing with symbolic inputs and embedded inputs. In other words, to produce a continuous representation from the symbolic ones, we just replace words by the clusters from where they belong (e.g., city_name, food, etc.) and keep the word if it does not belong to any of them. Thereafter, we produce embeddings by using a *Word2Vec* [70] model trained solely on the training set of a given corpus.

In order to have a strict comparison, the two different representations are then used as input for a classifier that is able to work with both of them. We use boosting over decision trees [57]. This algorithm is not specifically tailored for sequence labeling tasjs, but our current goal is to only compare the representations. Training is performed by learning an increasing number of weak classifiers over the training set of each respective dataset. The results are presented in Table 3.1 and they clearly show, on both datasets, that embedded representations improve the accuracy of the classifier. Moreover, we can observe in Figure 3.1 that embedded representations allow the classifier to converge significantly faster (in terms of F1-measure on the training set) than with symbolic representations, on both datasets. The classifier built on ATIS exhibits several drops in accuracy, as it can be seen in Figure 3.1a. Our explanation is that there are annotation errors in the ATIS dataset and that each drop corresponds to a rule created from this error by the classifier. As we can see, the embedded representation learned on the same corpus does not suffer from this drawback and appears to be noise robust. Annotation errors in ATIS are known since [88] who proposed a partially corrected ATIS version of the corpus, but some errors still remain [111, 88] show that in the previous noisy version, a basic HMM worked better than CRF because of their noise



(a) ATIS (b) MEDIA

Figure 3.1: F-measure according to the number of boosting iterations with symbolic and embedded features

Representation	Precision	Recall	F-measure
ATIS			
symbolic	93.00%	93.43%	93.21%
embedded	93.50%	94.54%	94.02%
MEDIA			
symbolic	71.09%	75.48 %	73.22%
embedded	73.61%	78.85%	76.14%

Table 3.1: Slot tagging performance obtained from symbolic and embedded representations using bonzaiboost on ATIS and MEDIA

resistance ability. After correction, every method benefited and gained up to 5 % absolute in accuracy, making CRF the best method. The fact that embedded representations perform better than the original symbolic representations under the same classifier indicates that the good results obtained on ATIS by different neural network architectures [68, 128, 129] are partially due to the representation itself.

It appears that using embedded representations in continuous representation spaces clearly brings advantages compared to using symbolic inputs. This advantage is due to the fact that embedded representations appear less sensitive to noise, avoiding the possibility for the classifier to build a very specific (and false) classification rule.

3.1.2 CRF and RNN Models

In this part, we extensively compare conditional random fields (CRFs) [54] to the recently proposed Elman and Jordan recurrent neural network models [68, 128], previously described in Section 2.1.3.1 of Chapter 2, on two datasets of different complexity to determine which model performs better and is more suited for the task of slot filling. In addition to conditional random fields and recurrent neural networks, for the sake of completeness, we also evaluate the previously used AdaBoost.MH [95] over bonsai trees [57]. Each of these algorithms is able to take as input an arbitrary set of features and observe features from preceding and

following positions of the sequence in an arbitrary window size. The main differences are as follows:

- AdaBoost.MH is a widely used classification algorithm that performs well on many different tasks. However, it is not dedicated at all to sequence labeling problems. Sequence tagging is done by successive and independent local decisions at each sequence position. Thus, this algorithm will give us a baseline to see improvements brought by the two next sequence adapted classification algorithms. We use the implementation described in [87].

- The standard behavior of a feedforward neural network is the same as for the previous algorithm: a succession of independent and local decisions. In recurrent neural networks, recurrence is added to allow the neural network to exhibit dynamic temporal behavior. In [68], they use the output of the neural network from the previous or future time step as a feature for the current neural network in the sequence. They proposed to use the hard predicted output or the output probabilities and test these solutions in both directions. In [128] they use as features in their recurrent neural network the output of the hidden layer of the previous time step. Despite these heuristics to trade off context information along the successive decision, no dependencies on target labels are explicitly modeled and no global decision is made. The recurrent neural network architectures tested are an Elman RNN and a Jordan RNN, both proposed by [68] and illustrated in Section 2.1.3.1 of Chapter 2. They have distributed their code based on the *Theano* library [5, 8].

- A conditional random field, unlike the previous algorithms is dedicated to sequence labelling. Target label dependencies are modeled under the Markov assumption (in order to remain tractable) and then a global decision on the sequence is made. However, popular and efficient implementations like the one we used [58] are capable of using solely symbolic features.

As stated before, all features have been extracted in windows of size $[-3, 3]$. This is a commonly used configuration that also gives the best results for both ATIS and MEDIA. Further increasing the window size didn't affect the result significantly. Smaller context window sizes would however decrease the performance.

For the symbolic feature representation, the feature set is composed of a bag of word / position pairs inside the windows. In case a word is found within the database of attributes (e.g., city_name), it is replaced with its corresponding entry prior to computing the representation. To obtain embedded representations, we used a skip-gram Word2Vec model [70] with hierarchical sampling, trained on the training corpus where words belonging to an attribute database were replaced by their corresponding attribute, in order to transfer this knowledge to the embedded representations. Only one embedding strategy is considered, since when fine-tuned, different word representations show very similar performances and provide comparable results [60]. This is also significantly cost-effective since just a few minutes are sufficient to compute the representations. Representations in a 100-dimensional space yielded very good results for all the tested classification algorithms. Further increasing the representation dimensionality did not result in a noticeable improvement of the results. This is the size we keep to do the algorithm comparison.

Algorithm	Info	Representation	Precision	Recall	F-measure	t_{train}
ATIS						
Bonzaiboost	100 iter	emb. (Word2Vec)	93.50%	94.54%	94.02%	20 m
Bonzaiboost	100 iter	symbolic	93.12%	92.82%	92.97%	3 m
CRF		symbolic	95.53%	94.92%	95.23%	6 m
Elman RNN	100 hdn	emb. (joint)	96.20%	96.12%	96.16%	1.5h
MEDIA						
Bonzaiboost	500 iter.	emb. (Word2Vec)	73.61%	78.85%	76.14%	2.5 h
Bonzaiboost	500 iter.	symbolic	71.09%	75.48 %	73.22%	34 m
CRF		**symbolic**	**87.70%**	**84.35%**	**86.00%**	**15 m**
Elman RNN	500 hdn	emb. (joint)	83.36%	80.22%	81.76%	31 h
Elman RNN	500 hdn	emb. (Word2Vec)	80.48%	83.46%	81.94%	22 h
Jordan RNN	500 hdn	emb.(joint)	82.76%	83.75%	83.25%	3.5 h
Jordan RNN	500 hdn	emb. (Word2Vec)	83.40%	82.90%	83.15%	3 h

Table 3.2: Slot filling performance of several learning algorithms on ATIS and MEDIA. In the 2nd column, "hdn" stands for the number of hidden neurons and "iter" for the number of iterations. The last column indicates the training time of each method.

In the RNN implementation [68], word embeddings are learned jointly with the final supervised task-specific classifier (RNN) by simply backpropagating the weights of the neural network during training and updating the representations within a lookup table. This has a small impact also on the speed of the overall training procedure. Database attributes have been integrated in order to provide a fair comparison. Recurrent neural networks have many crucial hyperparameters. We kept most of them fixed to the values proposed in [128]. We ran a 50 epochs learning and the best RNN configuration was selected according to its performance on the development set. On the other side, we kept the default parameters of wapiti. Bonzaiboost was ran with decision trees of depth 2 (max 4 leaves) according to [57]. For ATIS, we used the best data split reported in [128] while for MEDIA, the official split of the dataset has been used.

The three algorithms were ran on the slot extraction task for both databases: ATIS and MEDIA. Boosting and CRF implementations are multithreaded and were ran with 16 threads on a 2 Intel(R) Xeon(R) CPU X5560 @2.80GHz machine with 96 GB of RAM. The RNN GPU implementation was ran on an NVIDIA GeForce GT 750M 2048 MB graphic card. Performances were computed in terms of accuracy, precision, recall and F-measure, using the conlleval script[1]. Training times are also reported as a vague indicator of the complexities of the tested algorithms. Computations were made with different number of iterations (and hidden neurons for the case of RNNs) to ensure that the asimptote of the learning curve was reached. Table 3.2 reports these information for both ATIS and MEDIA.

Performance is very similar across classifiers on the ATIS dataset: from 93 % in F-measure for bonzaiboost (not dedicated to sequence labeling tasks and applied on symbolic representations) to 96 % for RNN. This result illustrates the fact that ATIS is not particularly challenging in terms of sequence classification. RNNs perform better (1 % absolute) than

[1] http://www.cnts.ua.ac.be/conll2000/chunking/output.html

CRF on ATIS. As pointed out in the Section 3.1.1, the representation used (symbolic for CRF and embedded for RNN) may explain the RNN gain. This result is also pointed out by the authors of [128].

On MEDIA, results are substantially different for each classifier. As expected, bonzaiboost, which is not dedicated at all to sequence labeling, produced the worst performance, around 76 %. RNNs follow with 83.25 % at the cost of high computational time. CRF, despite the fact that it is using less efficient symbolic representations, obtains 86 % with less computational cost (15min vs 3.5h). The Jordan variation of RNNs shows a less stable convergence. Elman RNNs had quite more stable convergence. Word embeddings learned in an unsupervised manner (word2vec) combined with an RNN perform similarly to word embeddings computed in a supervised manner, while learning the RNN classifier. However, precomputing the embeddings significantly decreases the time required for training an RNN classifier and helps the classifier converge faster. On the formulation side, CRF has the advantage to model explicitly the dependencies between target labels. To keep the CRF tractable, the linear chain CRF is widely used. This means that only dependencies between two adjacent labels are modeled. If we remove features related to these dependencies, CRF loses 6 % absolute in terms of F-measure. This result clearly indicates that the good performances of CRF derive from this dependency model.

Our results demonstrate that embedded representations allow for better accuracy and make the classification algorithm converge faster. Moreover, embedded representations decrease the possibility for a classifier to produce noise fitted decision rules and thus are more robust to noise than symbolic ones. Despite this conclusion, algorithms able to exploit them, like recurrent neural networks are not able to compete with conditional random fields. Although conditional random fields are trained solely on symbolic inputs, their ability to model output label dependencies appears crucial for the task. Conditional random fields with symbolic inputs thus remain the best classification algorithm for spoken language understanding in term of prediction (2.75 % absolute gain of F-measure in the challenging MEDIA corpus and a 16 % relative decrease of the error), simplicity (less hyperparameters) and rapidity (approximately 14 times faster in our experiments).

3.1.3 Embedding Output Label Dependencies

Up to this point we have shown that, while embedded representations allow for better accuracy, the ability of conditional random fields to model output label dependencies is crucial for obtaining state-of-the-art results in the task of slot filling. Recurrent neural networks do have a notion of the previous (and/or following, depending on the modeling direction) output or hidden state but this does not seem to be sufficient for obtaining state-of-the-art results comparable or better than conditional random fields.

In this section, we evaluate the idea, proposed by Marco Dinarelli [25], to use an embedding space for the output labels to model their dependencies and improve upon the problems recurrent neural networks face. We define a variation of the Jordan recurrent neural network architecture, previously described in Chapter 2, where the output labels are modeled in an embedding space that has been trained on sequences consisting of output labels. A single node of our proposed variant, named eJordan due to its additional embedding of the output labels, is illustrated in Figure 3.2. In this variant predicted labels are mapped into embeddings. However the embedding space of the input words is not the same as the

Method	**F1 (%)**	σ
ATIS		
Bidirectional Jordan	95.69	0.07
Bidirectional eJordan	95.74	0.02
CRF	95.23	0.00
MEDIA		
Bidirectional Jordan	86.15	0.09
Bidirectional eJordan	86.97	0.12
CRF	86.00	0.00

Table 3.3: Slot filling performance on ATIS and MEDIA of the classical Jordan architecture, our proposed eJordan modification and conditional random fields. The F1 measure and its respective standard deviation (computed over 10 runs) are reported.

embedding space of the output labels. Both were trained in an unsupervised manner on the training set of the evaluated datasets with *Word2Vec*. They were however trained separately. Word embeddings were trained on sequences of words (sentences) and output label embeddings were trained on sequences of output labels.

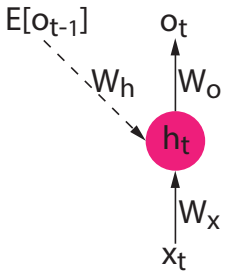


Figure 3.2: The proposed modification of the Jordan recurrent neural network architecture, named eJordan. The only difference between a normal Jordan architecture and the proposed is in the way the previous output o_{t-1} is passed to the current hidden layer. In our proposed architecture, the previous layer is first embedded in a representation space and $E[o_{t-1}]$ then passed to the hidden layer.

More formally, the architecture is given as follows:

$$\mathbf{h}_t = act_1(\mathbf{W}_h E[\mathbf{o}_{t-1}] + \mathbf{W}_x \mathbf{x}_t + \mathbf{b}_h) \tag{3.1}$$

$$\mathbf{o}_t = act_2(\mathbf{W}_o \mathbf{h}_t) \tag{3.2}$$

Where the only difference between the proposed variant and a standard Jordan RNN is that in our variant the label used as contextual information is first embedded and then passed back when computing the next recurrent iteration, as described by Equation 3.1. For this reason we name our variant *eJordan*, for *embedded Jordan* RNN.

We tested our proposed method, the classical Jordan architecture in a bidirectional sequence modeling setup (both forward and backwards) and report the results in Table 3.3, together with the previous results from conditional random fields. We see that just embedding the output labels offers enough additional information about their dependencies (just

like words, within their representation space, labels are grouped in clusters that are determined by their position in the sentence, context and cooccurrence) to improve over problem of output label dependencies and achieve state of the art today both on ATIS and MEDIA. After performing a single sided T-test, our performed eJordan architecture performed better than CRF with a significance of $\alpha = 0.001$ (obtained with a one-tailed t-test) both on ATIS and MEDIA.

3.2 Gated Recurrent Neural Networks

In this section, we evaluate more complex recurrent neural network architectures that contain gates that allow them to learn a model for dynamically retaining and clearing data, which consequentially allows for better modeling of longer sequences and should bring improvement over simple recurrent neural networks in the task of slot filling. We will start with a simple baseline consisting of a simple recurrent neural network architecture (a neural network consisting of cells / neurons containing just one recurrent connection), without discussing the already evaluated Elman and Jordan modifications that were presented in Section 3.1. We will then progress to evaluate different gated recurrent neural network architectures, as previously theoretically described in Section 2.1.3.2 of Chapter 2,: long short-term memory networks (LSTM) [40, 34] and novel gated recurrent units (GRU) [18]. In the end, we will conclude by analyzing each architecture in both their single-direction sequence modeling and bidirectional sequence modeling variations.

3.2.1 Simple Recurrent Neural Networks

Simple recurrent neural networks, as defined in Section 3.1, are neural networks where neurons have a recurrent weight pointing back at the same neuron and no additional weights other than the standard ones (the one weighting the input vector to the current neuron). In practice, such simple recurrent neural architectures have difficulties modeling long-term dependencies [7]. Gated recurrent networks such as LSTM and GRU networks, that we will discuss next, were introduced to improve upon this problem.

The parameters that worked best consist of an embedding size of 200 (the embeddings are learned jointly, while training the whole network), a context window of 11 (5 words before and 5 words after the current word) and an output size of the recurrent network of 200. Simple recurrent neural networks stopped improving with a smaller context window. However, a window of 11 did not make the results worse, so we kept the same window size over all experiments to have a more sensible comparison. The last fully-connected dense layer is always of a size equal to the number of output classes, after which a sigmoid activation layer follows. It was determined experimentally that a sigmoid activation layer performs better than layers with other common activation functions. We found that dropout of 50 % worked best. As illustrated in Table 3.4, this setup obtains an F-measure of 94.63 % on ATIS and 78.46 % on MEDIA.

3.2.2 Long Short-Term Memory Networks

Long short-term memory networks [40, 34] introduce a series of gates (input gate, forget gate and output gate) that help model the information retained by the recurrent network. LSTM networks have also been successfully used in spoken language understanding, either by themselves [129] or as encoder-decoder (sequence to sequence) architectures [53] that are more commonly used in machine translation tasks [4, 112].

We again used, an embedding size of 200 (the embeddings are learned jointly, while training the whole network), a context window of 11 and an output size of the recurrent network of 200, with dropout of 50 % and a sigmoid activation layer at the end. LSTM networks that model sequences in a forward direction obtained 95.12 % on ATIS and 81.54 % MEDIA. We already see that the improvement is clear on both datasets but less significant on ATIS, a problem we have already elaborated in Section 3.1.

3.2.3 Gated Recurrent Units

Gated recurrent units [18] are a recent variation of LSTM networks. They combine the forget and input gates into one update gate and merge the hidden state and cell state into one state. GRUs have been shown to perform better than regular LSTMs while also being faster due to a simpler architecture [20].

An embedding size of 200 with a context window of 11, an output size of the recurrent network of 200, with dropout of 50 % and a sigmoid activation layer were again used. The input was modeled sequentially in the forward direction, as in the previous subsections. Gated recurrent units performed better, even though they are simpler than LSTMs, and obtained 95.43 % on ATIS and 83.18 % on MEDIA.

3.2.4 Modeling Sequences in Both Directions

Recurrent neural networks typically model information solely in one direction, namely the forward one. In some cases, it's been shown that reversing the sequence can improve the performance of a recurrent network in machine translation applications [106]. It's thus best to model sequences in both directions (both $\dots x_{i-1}, x_i, x_{i+1} \dots$ and $\dots x_{i+1}, x_i, x_{i-1} \dots$). Modeling information in both directions can be done by implementing a bidirectional structure directly within the architecture of a recurrent neural network [97], or two recurrent neural networks working with opposing directions can be combined to achieve the same goal [135]. This last method is more common with complex recurrent neural networks and is also used in our work.

We implemented bidirectional LSTM and bidirectional GRU networks by duplicating the architecture and making one LSTM or GRU model the sequence in the opposing direction. Afterwards the outputs are combined by concatenation and fed to a fully-connected layer that generates an output-label prediction. All the other hyperparameters remain the same.

All the obtained results of the previously discussed methods are shown in Table 3.4 for easier comparison. On ATIS, we can state that bidirectional networks (bidirectional LSTMs and bidirectional GRUs) show improved performance over their monodirectional versions, although this statement is not very strong. On MEDIA, it's clear and more statistically significant that bidirectional gated recurrent networks work better than gated recurrent networks

Network Architecture	Accuracy	Precision	Recall	F1
ATIS				
RNN	97.71 (0.06)	94.02 (0.10)	95.26 (0.20)	94.63 (0.14)
LSTM	97.89 (0.04)	94.47 (0.18)	95.80 (0.19)	95.12 (0.17)
Bidirectional LSTM	97.91 (0.05)	94.61 (0.13)	95.86 (0.13)	95.23 (0.11)
GRU	97.95 (0.05)	94.72 (0.11)	96.14 (0.04)	95.43 (0.06)
Bidirectional GRU	98.00 (0.06)	94.86 (0.15)	96.21 (0.19)	95.53 (0.17)
MEDIA				
RNN	86.08 (0.25)	76.13 (0.67)	80.95 (0.23)	78.46 (0.45)
LSTM	87.80 (0.73)	80.49 (1.55)	82.61 (1.20)	81.54 (1.33)
Bidirectional LSTM	88.45 (0.05)	82.54 (0.85)	83.61 (0.22)	83.07 (0.37)
GRU	88.39 (0.16)	82.73 (0.56)	83.63 (0.38)	83.18 (0.47)
Bidirectional GRU	88.81 (0.09)	82.93 (0.42)	84.34 (0.33)	83.63 (0.16)

Table 3.4: Performances of the various recurrent architectures on ATIS and MEDIA. Averaged (over multiple runs) accuracy, precision, recall, F1 measure (%) and their respective standard deviations (in parenthesis).

that work solely in one direction: bidirectional LSTMs outperform LSTMs ($\alpha = 0.1$) and bidirectional GRUs outperform GRUs ($\alpha = 0.1$). We thus show that bidirectional sequence modeling is highly recommended, regardless of the architecture and that architecture-wise, gated recurrent networks perform better tha their non-gated counterparts with GRU networks achieving the best results.

3.3 Context Modeling

In the previous section, we described recurrent neural networks that learn how to dynamically retain or clear information stored in their internal states. However, for the task of slot filling, specific concepts that define the crucial part of the context needed to predict the current label are sometimes present further away in the sentence or even in another sentence that is part of the current dialog. For example, let's illustrate the case of the words *{l', hôtel}* in two phrases from the MEDIA dataset. The phrase *{je, souhaite, réserver, à,* ***l'****,* ***hôtel****, ibis} (I'd like to reserve at the Ibis hotel)* is labeled as *{B-command-tache, I-command-tache, I-command-tache,* ***B-hotel-marque****,* ***I-hotel-marque****, I-hotel-marque}* while the phrase *{savoir, s', il, y, a, un, restaurant, dans,* ***l'****,* ***hôtel****} (<missing begining> know if there is a restaurant inside the hotel)* is labeled as *{O, O, O, O, O, B-hotel-services, I-hotel-services, O,* ***B-lienRef-coRef****,* ***B-objetBD****}*. In this example, we focus on three labels: *hotel-marque* (hotel name/mark), *lienRef-CoRef* (referential/coreferential link) and *objetBD* (object). The difference between the first and the second sentence can be inferred by how far the dialog has progressed and what has already been mentioned so far. In this particular example, if the user is asking for specifics about a hotel, it means that there has probably been a previous mention of a hotel and that *"l'hôtel"* should be interpreted differently - as a reference to a previously mentioned object and not as a new hotel. Sometimes solely the current sentence suffices while in most cases, the knowledge of what has been previously mentioned improves the understanding of the current sentence.

We utilize a set of 37 word classes for ATIS and 19 word classes for MEDIA as relevant concepts within a context and we model a vector describing their presence from the beginning of the dialog to the current word of the current sentence. To illustrate, a few of word classes utilized for ATIS are: *{aircraft_code, airline_code, airline_name, airport_code, airport_name, city_name, class_type, cost_relative, country_name, day_name, ...}*. The presence of a word that belongs to one of those classes within the current dialog history (the sentences of the current dialog history, from the first sentence, until the current sentence) is encoded in a binary vector of length 37 or 19 for MEDIA and ATIS, respectively. In the MEDIA dataset, dialogues contain from 1 to 56 user sentences. ATIS does not provide a dialog so only the word classes from the current sentence are modeled.

Binary vectors containing information about the presence of word concepts are fed to a neural network in parallel with the context windows of the currently analyzed sentence. In the same way as in the previous setup without dialog awareness, words from the context window are first passed though an embedding layer after which two GRUs working in opposing directions follow. Their outputs are concatenated and dropout is applied. Dialog awareness vectors are instead passed through a dense, fully-connected layer of the same length as the input vectors. The two parts are then merged by a fully-connected dense layer of the size equal to the number of output labels and they are passed through a final activation function. Our proposed architecture is illustrated in Figure 3.3.

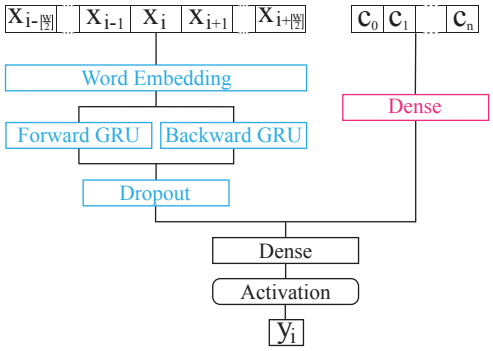


Figure 3.3: Our proposed architecture: a bidirectional GRU combined with a dialog aware fully-connected dense layer

Dialog aware bidirectional GRU networks were formed by adding a fully-connected dense layer that reacts to the vector describing which word classes were mentioned in the current dialog history (which is just the current sentence in the case of ATIS). The best results were obtained with only one fully-connected dense layer of size 37 (same size as the input - number of possible word classes) connecting the input to the merging layer.

On MEDIA combining word concepts from the dialog with bidirectional GRU networks further improves the results over bidirectional GRU networks that utilize solely the current sentence and gives 83.89 % ($\alpha = 0.27$), which is a significant improvement ($\alpha = 0.1$) over bidirectional GRUs without dialog awareness 83.63 % ($\alpha = 0.16$). On ATIS, the improvement is not significant and goes from 95.53 % ($\alpha = 0.17$) for bidirectional GRUs to 95.54 % ($\alpha = 0.16$) when the additional vector of key mentioned classes is added. Other that ATIS being a simple dataset where not many statistically significant conclusions can be brought, we believe this result is also influenced by the fact that ATIS does not posses dialogs and word

classes are modeled solely within the current phrase. We thus show that extending a gated recurrent neural architecture with a context modeling branch that incorporates the presence of relevant keywords can further improve the accuracy of a slot filling system and is crucial for datasets containing dialogs with multiple sentences.

3.4 Conclusion

In the first part of this chapter, we show that, although embedded representations bring improvement over symbolic ones, the ability of conditional random fields to model output label dependencies is crucial. In order to evaluate possible gain sources in recurrent neural networks, we compared symbolic and embedded, continuous word representations for spoken language understanding with a classification algorithm able to use both. Our results demonstrate that the latter allows for better accuracy and makes the classification algorithm to converge faster. Moreover, continuous representations decrease the possibility for a classifier to produce noise fitted decision rules and thus are more robust to noise than symbolic ones. Although conditional random fields are trained solely on symbolic features, their ability to model output label dependencies appears crucial for the task. We thus proposed a modification of the Jordan architecture that models the output labels into a representation space, learned in an unsupervised manner on sentences consisting of the output labels from a training set. Our proposed architecture improved not only over the classical Jordan architecture but also over conditional random fields and achieved the state of the art today on both ATIS and MEDIA.

In the second part of this chapter, we evaluated gated recurrent neural networks and extending the architecture in such a way that key conceptual concepts from the dialog can be incorporated. We evaluated different gated recurrent neural networks and we analyzed the possibility of modeling key concepts within the dialog. We show that gated recurrent neural networks, known for better long dependency modeling, clearly outperform simple recurrent neural networks. Within gated recurrent neural networks, we show that gated recurrent unit based networks outperform long short-term memory based networks. Gated recurrent networks model information in one direction. Modeling information in both direction by combining two networks with opposing direction improves performance, as demonstrated with both bidirectional LSTM networks and bidirectional GRU networks. Finally, we show that adding information about the presence of specific word classes within the current dialog history further improves the performance of the previously best-performing bidirectional GRU networks. Unfortunately, simple CRF methods still slightly outperform RNN methods [118]. We believe this is due to better target dependency modeling that CRF offers. However, RNNs represent a competitive framework that might offer easier extensions such as attention models implemented over the dialog history.

We believe that there is not much improvement left to be done with architectures that model output labels independently (one by one) and utilize solely the current sentence. As shown, improvement can be achieved by integrating distant dependencies that are part of the dialog but are not necessarily part of the current sentence. In our opinion, future work should address means of incorporating knowledge from the entire dialog, either by engineering relevant features or by deploying appropriate attention models. As a consequence, experiments performed on datasets more complex than ATIS are also required.

Chapter 4

Action Forecasting

Contents

There is an inherent need for machines to have a notion of how entities within their environment behave and to anticipate changes in the near future. Machines typically have a response time $\Delta t_{response}$. Being able to anticipate the near future allows them to correct for their inherent delay and to plan accordingly. Anticipating the near future is especially useful in robotics, where artificial systems have to interact with their environment in real time. This is however a difficult task since even with recent advances in deep and reinforcement learning, machines still do not possess complex knowledge of the world and are rather adapted to specific narrow tasks. If we limit the task to anticipating future appearance of video frames, machines have a slight advantage due to the vast collection of unlabeled videos available today which is perfectly suited for unsupervised learning methods. To anticipate future appearance based on current visual information, a machine needs to successfully be able to recognize entities and their parts, as well as to develop an internal representation of how does the movement happen in regards to time.

We start from a given input video frame and aim to predict a future video frame at a given temporal distance, Δt away from the input frame. We achieve this by conditioning our video frame prediction on an input time-indicating variable and we are able to perform a one-step prediction of the future video frame that is temporally further away from the input given frame. Therefore, in this thesis we propose one-step, long-term video frame prediction. This is beneficial both in terms of computational efficiency, and for not having to be concerned with the propagation and accumulation of prediction errors, as in the case of sequential/iterative prediction.

Our work falls into the encoder decoder category of neural architectures, where a current image is presented as input and an image resembling the anticipated future is provided as output. Our proposed method consists of an encoding CNN, a decoding CNN and a separate branch, parallel to the encoder, that models time.

4.1 Overview of Possible Approaches

In the context of action prediction, it has been shown that it is possible to use high level embeddings to anticipate future actions up to one second before they begin [113]. Predicting the future event by retrieving similar videos and transferring this information, is proposed in [132]. In [56] a hierarchical representation is used for predicting future actions. Predicting a future activity based on analyzing object trajectories is proposed in [49]. In [42], the authors forecast human interaction by relying on body-pose trajectories. In the context of robotics, in [50] human activities are anticipated by considering the object affordances. Unlike these works, rather than predicting actions, we focus on predicting a single video frame at a given future temporal displacement from a given input video frame.

Anticipating future movement in the spatial domain as closely as possible to the real movement has also been previously considered. For this case, the methods start from an input image at the current time stamp and predict optical flow (OF) at the next timestep. In [63] images are aligned to their nearest neighbour in a database and the motion prediction is obtained by transferring the motion from the nearest neighbor to the input image. In [83], structured random forests are used to predict OF vectors at the next timestep. In [82], the use of LSTM is evaluated for predicting Eulerian future motion. A custom deep convolutional neural network is proposed in [121] towards future OF prediction. Rather than predicting the motion at a future moment in time, in [123] the authors propose to predict motion trajectories using variational autoencoders. This is similar to predicting OF, but given the temporal consistency of the trajectories it offers greater accuracy. Dissimilar to these methods, we aim to predict the video appearance information at a given future temporal displacement, from an input video frame. Predicting appearance rather than motion is beneficial as the predicted outcome is spatially coherent.

Focusing on the object motion as given by their dynamics in real world, is proposed in [72], by relying on Newtonian physical laws. In [31], the future location of objects is predicted by learning from synthetic abstract data. This can be seen somewhat related to learning to predict OF, which is also an indicator of displacement. Unlike these methods, we aim to predict the appearance of a future video frame given an input video frame and a desired temporal difference.

One intuitive trend towards predicting future information is predicting future appearance. In [122], the authors propose to predict both appearance and motion for street scenes using top cameras. Predicting patch-based future video appearance, is proposed in [86], by relying on large visual dictionaries. Similar to these methods, we also aim at predicting the appearance of future video frames, however we condition our prediction on a time parameter that allows us to perform the prediction efficiently, in one step.

More recent methods rely on convolutional neural networks towards predicting possible video frames. Rather than predicting future appearance from input appearance information, hallucinating possible images has been a recent focus. The novel work in [114] relies on the

generative adversarial network model [84] to create not only the appearance of an image but also the possible future motion. This is done using spatio-temporal convolutions that discriminate between foreground and background. Similarly, in [93] a temporal generative neural network is proposed towards generating more robust videos. These generative models can be conditioned to generate feasible outputs given a specific conditioning input [89, 123]. Dissimilar to them, we rely on an autoencoding model. Autoencoding methods try to encode the current image in a representation space that is suitable for learning appearance and motion, and decode such representations to retrieve the anticipated future, either as an image or optical flow/trajectories. We propose to use video frame appearance towards predicting future video frames, conditioned on a given time indicator.

Related to predicting future appearance, the recent work in [75, 76] propose predicting future image pixels conditioned on all previous seen pixels — possible image completions from a set of initial pixels. Unlike these methods, we aim to predict complete future images from a provided input image and a provided temporal displacement.

Similar to transferring the optical flow vectors between images, as considered in [63], appearance transfer has also been considered. Work such as [32, 46, 91] focuses on the task of artistic style transfer from a given input image to another image or video. Unlike these methods, we do not transfer a given appearance but rather predict a future frame appearance. We do so by conditioning on a parameter indicating the desired time displacement between the input frame and the predicted frame.

4.2 Architectures

The simplest and most straightforward method for generating predictions at a temporal distance Δt is by using an architecture that is trained to predict at a temporal distance Δt_p and then iteratively use the predicted image as input to the same architecture to predict at $2\Delta t_p$, $3\Delta t_p$ and so on, until we reach $k\Delta t_p = \Delta t$. This is illustrated in Figure 4.1. The downside of this approach is the discretization of the possible temporal displacements, which is usually bound to the discretization determined by the framerate of the videos used for training.

To tackle the problem of discretized possible temporal distances between the input image and the obtained prediction, we extended the architecture with an additional branch of fully-connected layers that models time, as illustrated in Figure 4.2. The encoder has two separate branches, one to receive the input image, and one to receive the desired temporal displacement Δt of the prediction. The decoder then takes the input from the encoder and generates a feasible prediction for the given input image and the desired temporal displacement.

The network receives as inputs an image and a variable Δt, $\Delta t \in \mathbb{R}^+$, indicating the time difference from the time of the provided image to the time of the desired prediction. The network predicts an image at the anticipated future time $t_0 + \Delta t$. We use a similar architecture to the one proposed in [109]. However, while their architecture is made to encode RGB images and a continuous angle variable to produce RGBD as output, our architecture is designed to take as input a monochromatic image and a continuous time variable, Δt, and to produce a monochromatic image as output. More specifically, the architecture consist of the following:

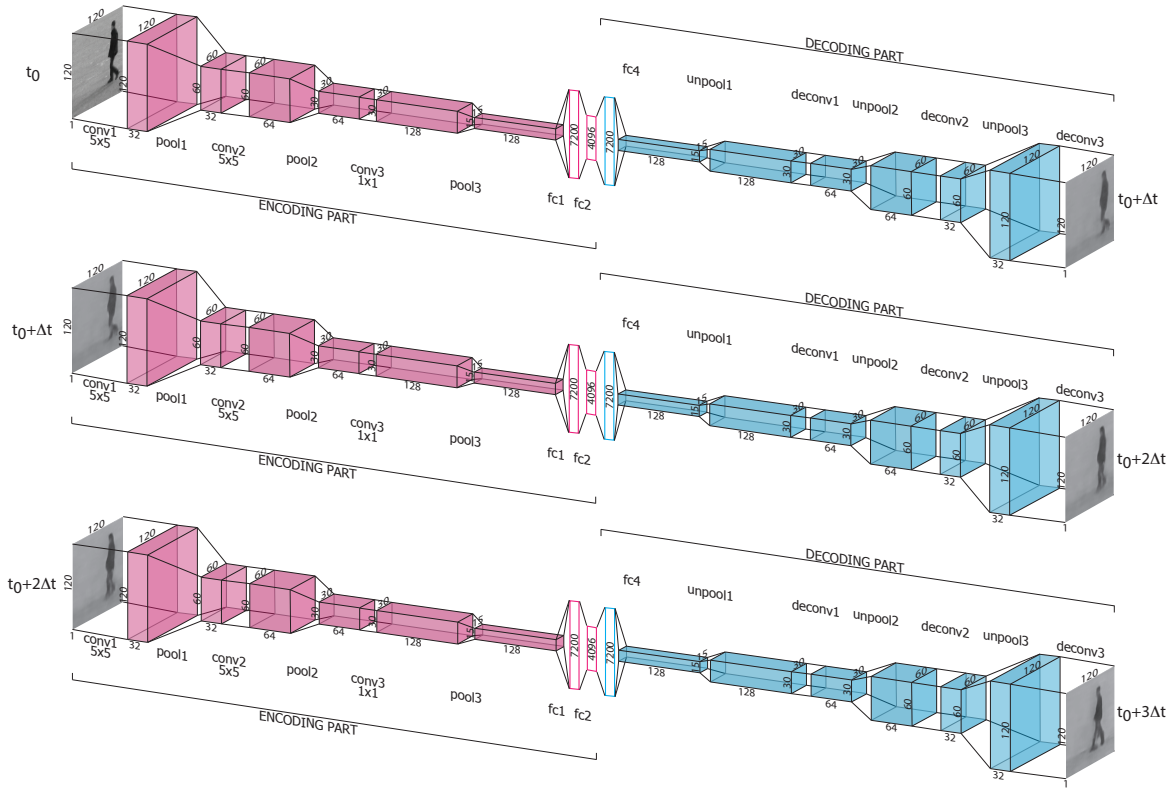


Figure 4.1: Iterative forecasting: the original input image at time t_0 is used to generate a prediction at time $t_0 + \Delta t_p$, which is then presented as input to the same architecture to generate a prediction for $t_0 + 2\Delta t_p$ and so on.

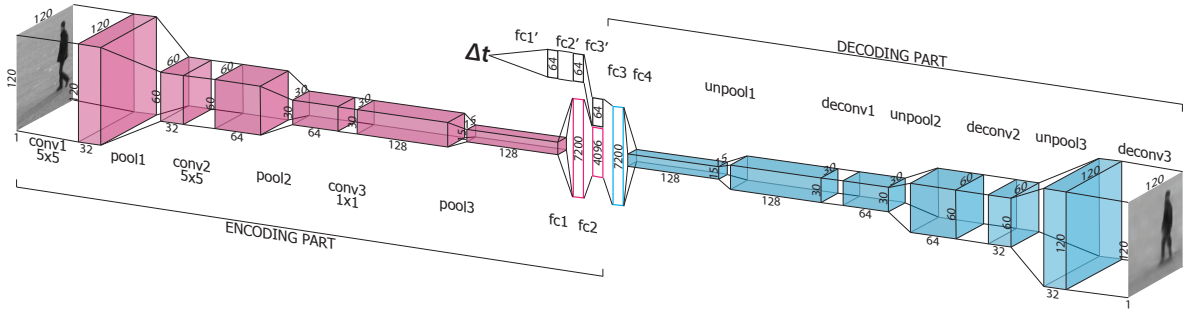


Figure 4.2: Our proposed architecture consists of two parts: i) an encoder part consisting of two branches: the first one taking the current image and the second one taking an arbitrary time difference t to the desired prediction and ii) a decoder part that generates an image, as anticipated, at the desired input time difference.

- *an encoding part* composed of two branches:
  - *an image encoding branch* defined by 3 convolutional layers, 3 pooling layers and 2 fully-connected layers at the end;
  - *a time encoding branch* consisting of 3 fully-connected layers.

  The final layers of the two branches are concatenated together, forming a new representation that is then provided to the decoding part.
- *a decoding part* composed of 2 fully-connected layers, 3 "unpooling" (upscaling) layers, and 3 "deconvolutional" (transpose convolutional) layers.

The input time variable is continuous and allows for appearance anticipations at arbitrary time differences. Possible alternatives of the proposed architecture could include encoded time inputs (e.g., multiple input neurons) or a continuous time variable followed by an embedding layer (e.g., lookup table). The downside of these approaches would be the discretization of the time input.

Training is performed by presenting batches of $\{I_x, \Delta t, I_y\}$ tuples, where I_x represents an input image at current relative time t_0, Δt represents a continuous variable indicating the time difference to the future video frame and I_y represents the actual video frame at $t_0 + \Delta t$.

Predictions are obtained in one step. For every input image I_x and continuous time difference variable Δt, a $\{I, \Delta t\}$ pair is given to the network and an image representing the appearance anticipation I_y after a time interval Δt is directly obtained as output.

4.3 Experiments

We evaluate our method by generating multiple images of anticipated future appearances and comparing them both visually and through MSE (Mean Squared Error) with the true future frames, as well as to a CNN baseline method that sequentially predicts the future video frame. For the baseline method, we use a CNN encoder-decoder architecture that does not have a notion of time and is used in an iterative manner to produce anticipated futures at $k\Delta t$ $(k = 1, 2, ...)$ temporal displacements.

4.3.1 Experimental Setup

To test the proposed architecture, we implemented it by using the *TensorFlow* [1] framework. We use the Adam optimizer [48], with L_2 loss and dropout rate set to 80 % for training. We argue that the type of action can be easily automatically detected and is better incorporated by training a network per action category. Thus, we opt to perform separate preliminary experiments for each action instead of training one heavy network to anticipate video frames corresponding to all the different possible actions. Training is performed up to 500000 epochs with randomized minibatches consisting of 16 samples where each sample contains one input image at current relative time $t_0 = 0$, a temporal displacement Δt $(\Delta t < 200ms)$ and the real frame at the desired temporal displacement Δt. We do not use early stopping and we ran each experiment for the full number of epochs. On a *Titan X* GPU, training took approximately 16 hours with, on average, about 100,000 training samples (varying in each action category).

Given that the input, and thus also the output, image size is $120 \times 120 \times 1$ (120×120 grayscale images), in our encoder part, we stack convolutional and pooling layers that yield consecutive feature maps of the following decreasing sizes: 120×120, 60×60, 30×30 and 15×15 with an increasing number of feature maps per layer, namely 32, 64 and 128 respectively. Fully-connected layers of sizes 7200 and 4096 follow. The separated branch of the encoder that models time consists of 4 fully-connected layers of size 64, where the last layer is concatenated to the fully-connected layer on top of the convolutional neural networks. This yields an embedding of size 4160 that is presented to the decoder. Kernel sizes used for the convolutional operations start at 5×5 in the first layers and decrease to 2×2 and 1×1 in the deeper layers of the encoder. For the decoder, the kernel sizes are ordered in the opposite direction.

The decoder consists of interchanging "unpooling" (upscaling) and "deconvolutiton" (transpose convolution) layers, yielding feature maps of the same sizes as the image-encoding branch of the encoder, only in the opposing direction. For simplicity, we implement pooling as a 2D convolution and unpooling as a 2D transpose convolution. It is worth noting that sometimes pooling/unpooling layers are completely omitted [104, 109] in similar encoder-decoder CNN architectures with no significant impact on performance. We decided to keep them as a regularization term given that our input and output images differ less and have a more similar appearance than in the case of rotated images [109].

We use the KTH human action recognition dataset [96] for evaluating our proposed method. The dataset consists of 6 different human actions, namely walking, jogging, running, hand-clapping, hand-waving and boxing. Each action is performed by 25 actors. There are 4 video recordings for each action performed by each actor. Inside every video recording, the action is performed multiple times and information about the time when each action starts and ends is provided with the dataset.

To evaluate our proposed method properly, we randomly split the dataset by actors, in a training set, with 80 % of the actors, and a testing set, with 20 % of the actors. By doing so, we ensure that no actor is present in both the training and the testing split and that the network can generalize well with different looking people and does not overfit to specific characteristics of specific actors. The dataset provides video sections of each motion in different directions - e.g., walking from right to left and from left to right. This provides a good setup to check if the network is able to understand human poses and locations, and correctly anticipate the direction of movement. The dataset was processed as follows: frames of original size 160×120 were cropped to 120×120 and the starting/ending times of each action are adjusted accordingly to match the new cropped area. Time was estimated based on the video framerate and the respective frame numbers.

4.3.2 Experimental Results

Our method is evaluated as follows: an image at a considered time, $t_0 = 0$ and a time difference Δt is given as input. The provided output represents the anticipated future frame at time $t_0 + \Delta t$, where Δt represents the number of milliseconds after the provided image.

The sequential encoder-decoder baseline method is evaluated by presenting solely an image, considered at time $t_0 = 0$ and expecting an image anticipating the future at $t_0 + \Delta t_b$ as output. This image is then fed back into the network in order to produce an anticipation of the future at time $t_0 + k\Delta t_b$, $k = 1, 2, 3, ...$

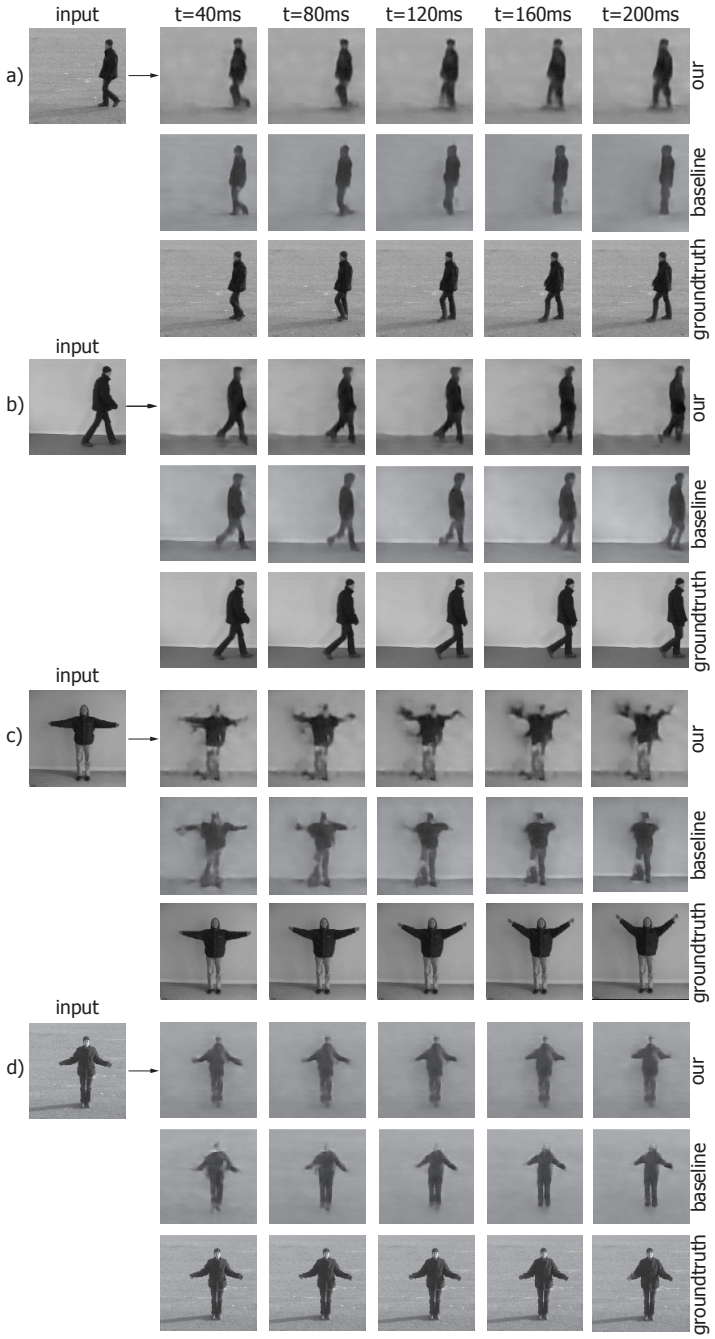


Figure 4.3: Comparison of predictions for a) a person walking to the left, b) a person walking to the right, c) a person waving with their hands and d) a person slowly clapping with their hands. Given an input picture (on the left) and a time interval (different columns) anticipated future motions are presented for our proposed method and for the baseline convolutional encoder-decoder. The third set of images in each group present the actual future — the groundtruth.

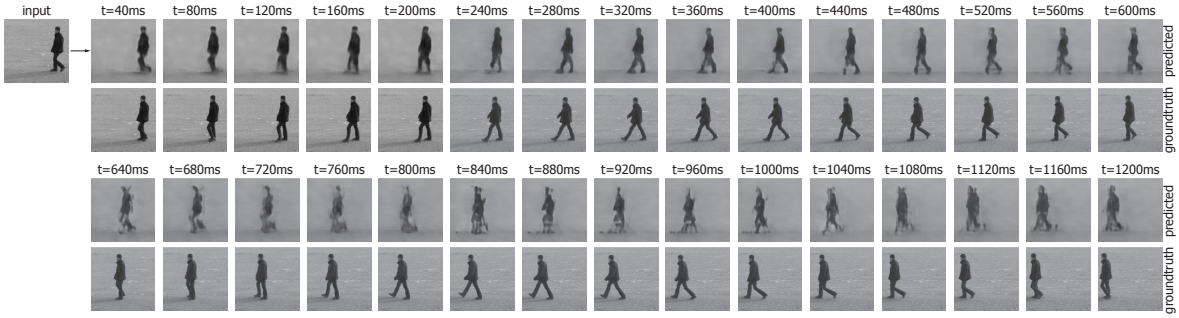


Figure 4.4: Long distance predictions. For larger temporal displacements artifacting becomes visible. The anticipated location of the person begins to differ from the groundtruth for even larger temporal differences, towards the end of the total motion duration.

For simplicity, we consider $t_0 = 0ms$ and refer to Δt as simply t. It is important to note that our method models time as a continuous variable. This enables the model to predict future appearances at previously unseen time intervals, as seen in Figure 4.6. The model is trained on temporal displacements defined by the framerate of the training videos. Due to the continuity of the temporal variable, it can successfully generate predictions for: i) temporal displacements found in the videos (e.g., *t={40ms, 80ms, 120ms, 160ms, 200ms}*, ii) unseen temporal displacement within the values found in the training videos (e.g., *t={60ms, 100ms, 140ms, 180ms}*) and iii) unseen temporal displacement after the maximal value encounter during training (e.g., *t=220ms*).

Since both the baseline method and the groundtruth are quantized by the video framerate, the images displayed in Figure 4.3 are all images at intervals of 40 ms (derived from a framerate of 25fps) for a fair and exact comparison. Figure 4.3 a) illustrates the case of a person moving from right to left, from the camera viewpoint, at walking speed. Despite the blurring, especially around the left leg when asked to predict for $t = 120ms$, it can be noticed that our proposed network correctly estimated the location of the person and positioning of body parts. For each time difference, the body-part predictions are realistic, as well as the displacement of the whole person, which matches the groundtruth displacement.

Figure 4.3 b) again illustrates a person walking, this time left to right. Our proposed network correctly localized the person and the body parts. The network is able to estimate the body pose and thus the direction of movement. Our network correctly predicts the displacement of the person to the right for any given time difference, from just the single input image.

The network is able to capture the characteristics of the human gait, as it predicts correctly the alternation in the position of the legs. The anticipated future frame is realistic but not always perfect, as it is hard to perfectly estimate walking velocity solely from one static image. This can be seen at $t = 200ms$ in Figure 4.3 b). Our network predicts one leg to further behind while the actor, as seen in the groundtruth, was moving slightly faster and moved the leg past the knee of the other leg.

Our proposed network is able to learn an internal representation that is capable of encoding the stance of the person such that it correctly predicts the location of the person, as well as to anticipate their new body pose after a deliberate temporal displacement. The baseline network does not have a notion of time and therefore relies on iterative predictions.

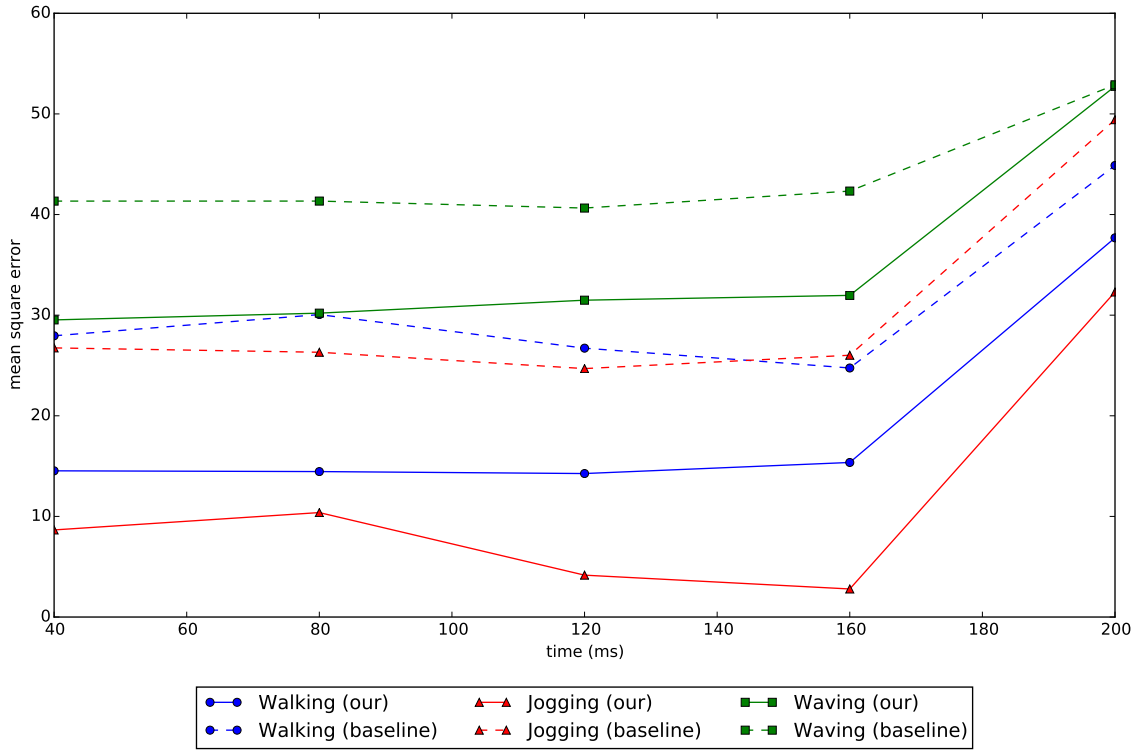


Figure 4.5: Mean squared error (MSE) over time for certain actions (walking, jogging, waving) for our proposed method and for the convolutional encoder-decoder baseline.

Time is quantized and the network is trained to generate an anticipated image at time $t + \Delta t$, given an image at time $t = 0$. After that, the process is repeated iteratively, which affects the performance. Figure 4.3 shows that the baseline network looses the ability to correctly anticipate body movement after some time. This can be best seen in Figure 4.3 a) where the baseline network correctly predicts the position of the legs up to $t = 80ms$. After that, the network predicts correctly the global displacement of the person in the correct direction, but body part movements are not anticipated correctly. At $t > 160ms$ the baseline encoder-decoder network shows a big loss of details, enough to cause its inability to correctly model body movement. Therefore, it displays fused legs where they should be separated, as part of the next step the actor is making. Our proposed architecture correctly models both global person displacement and body pose, even at $t = 200ms$.

Figure 4.3 c) displays an actor handwaving. The proposed network successfully predicts upward movement of the arms and generates images accordingly. In this case however, more artifacts are noticeable. The bidirectional motion of hands during handwaving is ambiguous, as the hand pose does not affect other body parts such as head positioning, or legs.

It is important to note that although every future anticipation is independent from each other they are all consistent: i.e., it does not happen that the network predicts one movement for t_1 and a different movement for t_2 that is inconsistent with it. This is a strong indicator that the network learns an embedding of appearance changes over time, the necessary filters to react to relevant image areas, and to synthesize correct future anticipations.

	Mean Squared Error (%)	
Action	**Baseline**	**Proposed Method**
Jogging	30.64	11.66
Running	40.88	17.35
Walking	30.87	19.26
Hand-clapping	43.23	33.93
Hand-waving	43.71	35.19
Boxing	46.22	37.71
Mean	39.26	25.85

Table 4.1: Averaged MSE, over multiple time differences and multiple predictions, on the different action categories of KTH. We compare our method with the baseline convolutional encoder-decoder and show that our method on average performs better than the baseline method in terms of MSE.

However, our proposed model is limited by the total temporal displacement t. For very large time displacements, we expect our frame predictions to deteriorate. This is emphasized in long-term anticipations, as illustrated in Figure 4.4. The smaller the temporal displacement t, the better the prediction is and the lower the MSE score, when compared to the real future frame. In this work, we do not check the limits of a maximum feasible time difference t, after which our proposed method would provide unsatisfactory results. However, as seen both from the illustrations in Figure 4.3 and the graphs in Figure 4.5, our network behaves better with respect to increasing time displacements than the encoder-decoder baseline network. This is supported by the network's ability to predict future video frames at arbitrary future times directly, without having to go through iterative steps that accumulate prediction error.

As expected, not every action is equally challenging for the proposed architecture. Table 4.1 illustrates MSE scores averaged over multiple time differences, t, and for different predictions from the KTH test set. MSE scores were computed on dilated edges of the groundtruth images to only analyze the part around the person and remove the influence of accumulated variations of the background. A Canny edge detector was used on the groundtruth images. The edges were dilated by 11 pixels and used as a mask for both the groundtruth image and the predicted image. MSE values were computed solely on the masked areas.

We compare our proposed method with the baseline CNN encoder-decoder architecture. It is worth noticing that the MSE does not strictly correlate with qualitative visual inspection. For example, on average, running seems to perform reasonably well, and moreover it outperforms hand-waving, hand-clapping, boxing and even walking. Yet, this is not the case as predictions for running, at the framerate available in the KTH dataset, generate a considerable loss of details and artifacts, as visible in Figure 4.7 d). These artifacts are not as prominent in the other, less well-performing action categories, in terms of MSE scores. The average MSE scores, given in Table 4.1, show that our proposed method outperforms the encoder-decoder CNN baseline by a margin of 13.41, on average, which is expected due to the iterative process of the baseline network.

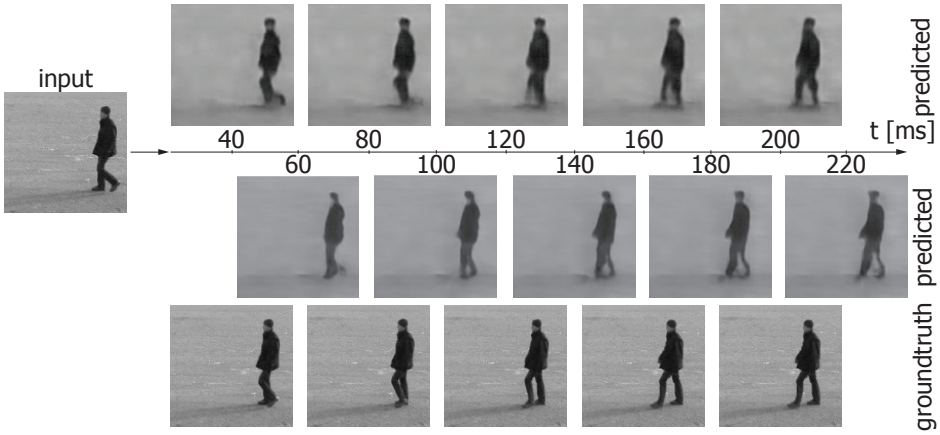


Figure 4.6: Prediction of seen and unseen temporal displacements. The networks is trained on temporal displacements dictated by the training set video framerate. However, predictions are possible both for seen (1st row, $t = 40 \times k$ ms) and for previously unseen temporal displacements (2nd row, $60 + 40 \times k$ ms).

4.3.3 Ambiguities and Downsides

As MSE values grouped by different actions indicate, not every action is equally challenging for our proposed method to be anticipated. However, there are a few key factors that make prediction more difficult and cause either the creation of artifacts or loss of details in the generated future frames.

4.3.3.1 Human Pose Ambiguities

Ambiguities in body-pose happen when the subject is in a pose that does not display inherent information about the movement of the subject in question. A typical example would be when a person is waving, moving their arms up and down, and an image with the arms at a near horizontal position is fed to the network as input. This can result in small artifacts, as visible in Figure 4.3 c) where for larger time intervals t, although the network is generating upward arm movement, there are visible artifacts that are part of a downward arm movement. A more extreme case is shown in Figure 4.7 a) where not only does the network predict the movement wrong, upward instead of downward, but it also generates a lot of artifacts with a significant loss of details that increases with the time difference, t.

4.3.3.2 Fast Movement

Fast movement causes extreme loss of details when the videos provided for training do not offer a high-enough framerate. In other words, this case happens when the visual difference between two consecutive frames during training is substantial — large global displacement and a body pose change that are too large. Examples of this can be seen in Figures 4.7 b) and c) where the increased speed in jogging and an even more increased speed in running generate significant loss of details. It is important to emphasize that although our proposed architecture can generate predictions at arbitrary time intervals t, the network is still trained

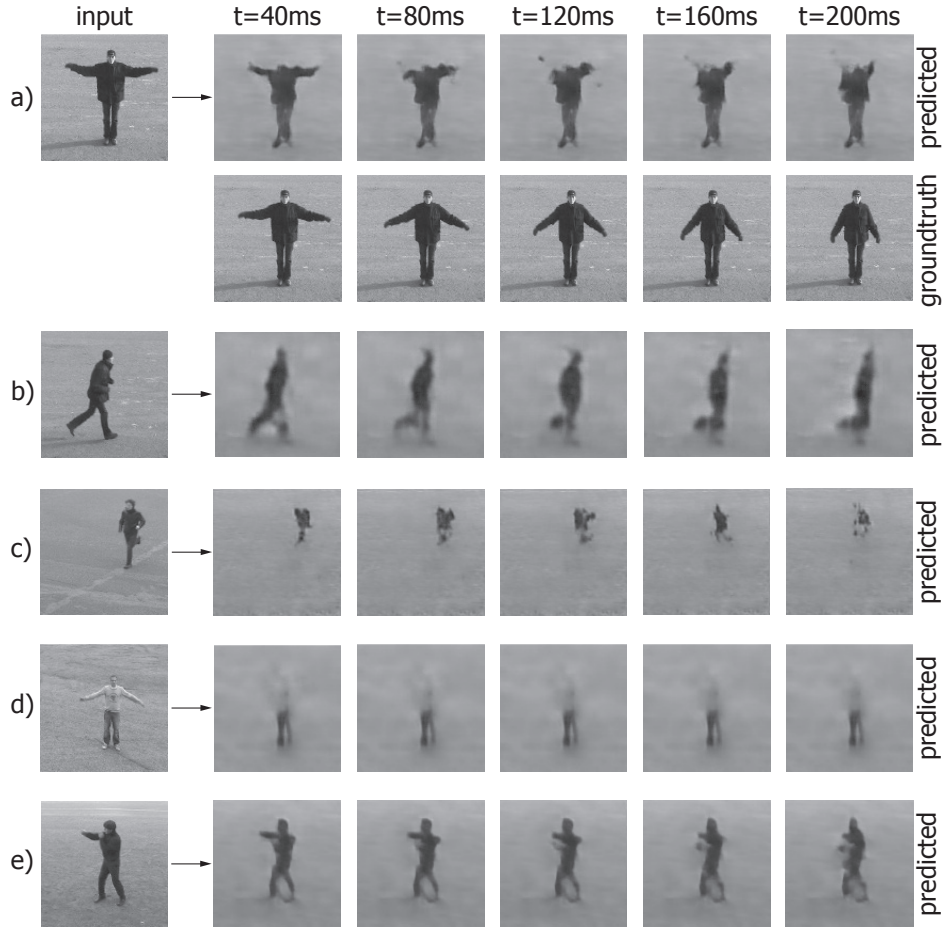


Figure 4.7: Examples of poor performing future anticipations: a) loss of details in waving, b) loss of details in jogging, c) extreme loss of details in running, d) loss of details with low contrast and e) artifacts in boxing.

on discretized time intervals derived from the videos — intervals that might not be small enough for the network to learn a good motion model. We believe this causes the loss of details and artifacts, and using higher framerate videos during training would alleviate this.

4.3.3.3 Insufficient Foreground/Background Contrast

Decreased contrast between the subject and the background describes a case where the intensity values corresponding to the subject are similar to the ones of the background. This leads to an automatic decrease of MSE values and a more difficult convergence of the network for such cases, which leads to less adaptation and thus to loss of details and artifacts. This can be seen in Figure 4.7 d). Such effect would be less prominent in case of modeling a network using color images.

4.3.3.4 Excessive Localization of Movements

Excessive localization of movements happens when the movements of the subject are small and localized. A typical example is provided by the boxing action, as presented in the KTH dataset. Since the hand movement is close to the face and just the hand gets sporadically extended — not a considerable change given the resolution of the images — the network has more difficulties in tackling this. Despite the network predicting feasible movement, often artifacts appear for bigger time intervals t, as visible in Figure 4.7 e).

Although the previously enumerated cases can lead our proposed architecture to predict that display loss of details and artifacts, most can be tackled and removed if necessary by either increasing the framerate, the resolution of the training videos, or using RGB information. The most difficult factor to overcome is human pose ambiguity. We believe this is a hard problem for our proposed architecture to manage.

4.4 Conclusion

We have successfully shown that a convolutional encoder-decoder network with an added fully-connected branch that models time can accurately generate latent representations that include both information about the location of a person and their stance, as well as temporal information. This allows the decoder part of the network to synthesize anticipations that correctly predict not only the displacement in the correct direction of a person performing an action but also correctly animate their stance for the correct amount that is time dependent. Not only our proposed method allows predictions in one step but it also provides better predictions than an iterative encoder - decoder architecture that tends to degrade after a few iterations. In other words, we not only improved in terms of prediction speed but also in terms of accuracy.

This is a novel notion that can be extended further and that yields high quality anticipations of future video frames for arbitrary temporal displacements, without having to explicitly model the time period in between the provided input video frame and the requested anticipation.

Chapter 5

Multimodal Continuous Representation Spaces

Contents

In the previous chapters we explored different scenarios where only one modality is used (e.g., only speech or only images). The setup discussed in Chapter 4 can, however, also been viewed in a semi multimodal way, where the image input represents one modality and the temporal input another modality. In this chapter, we introduce methods to deal with multimodal data that we later evaluate in the task of video hyperlinking. After providing definitions and giving an introduction of existing multimodal autoencoders we proceed to lying the theoretical grounds of two methods that we propose: i) bidirectional deep neural networks as an improvement over multimodal autoencoders that take initially disjoint multimodal representations and provide superior multimodal representations, and ii) a method to use conditional generative adversarial networks to perform multimodal fusion but also obtain visualizations of the learned model directly into the image domain. These methods will further be evaluated in the task of video hyperlinking in the following chapters.

5.1 Dealing with Multiple Modalities

We define a "modality" as a data collection aggregated by an acquisition framework [55]. Two typical acquisition frameworks are: i) image acquisition (performed typically with

CMOS or CCD sensors) that sense the world and output a discrete representation in the image space (images) or a temporal image space (videos), and ii) sound acquisition (performed typically with different types of microphones and analog to digital converters). Each modality can then be transformed and represented in different ways. Raw audio containing speech can be automatically transcribed and used as text or it can be represented with i-vectors [24] and used for speaker recognition / identification. Video keyframes can be described with human-understandable ImageNet concepts [92] or with descriptors obtained with convolutional neural networks [98, 51].

Figure 5.1 illustrates two modalities, an audio modality and a visual modality, and the three different levels at which they can be represented. In all generality, there are three main distinct data presentation levels at which each modality can be represented:

- **The original domain** - is the domain where the data is represented after acquisition and discretization. This could be a discrete 1D signal for audio or a sequence of RBG tensors for a visual modality.
- **The concept space** - is a space that describes the original space with key concepts or keywords. Such a space is typically not very practical for machine learning applications but is usually human-interpretable and can provide a very simple summary of what has been acquired in the original space. Typical examples are automatic transcripts for speech from an audio source or even just extracting main keywords from the transcripts and ImageNet concepts describing the content of a video keyframe.
- **The representation space** - is the most useful space for machine learning applications. It can be a discrete representation space obtained from the concept space (e.g., bag-of-words representation of the automatic transcripts) or a continuous representation space (e.g., Word2Vec). In deep learning applications, continuous representation spaces are typically used and achieve state-of-the-art performances. Representation spaces, especially continuous representation spaces are less human interpretable than concept spaces but they are very useful for machine learning applications and offer nice properties [70]. Useful properties of continuous representation spaces (or latent spaces if used solely within an architecture) can also be seen in the original domain by synthesis with generative adversarial networks [35, 85, 43].

When dealing with data that contain more than one modality (e.g., captioned images, videos, transcribed audio, etc.), there is an inherent need to combine them. There are two distinct approaches that integrate each modality:

- **Multimodal fusion** - is the approach of combining the representations each of modality into a new representation that contains the unified, but not necessarily disjoint, information of both input modalities [66, 15, 73]. Typical examples of such approaches include multimodal retrieval of personal photos using both visual representations and text representations of given captions [66], using both visual and speech representations in video retrieval [73] and many other tasks where multiple modalities are available.
- **Crossmodal translation** - on the other hand, translates from one modality to the other, without combining the information of both input modalities [30, 117]. Such approaches

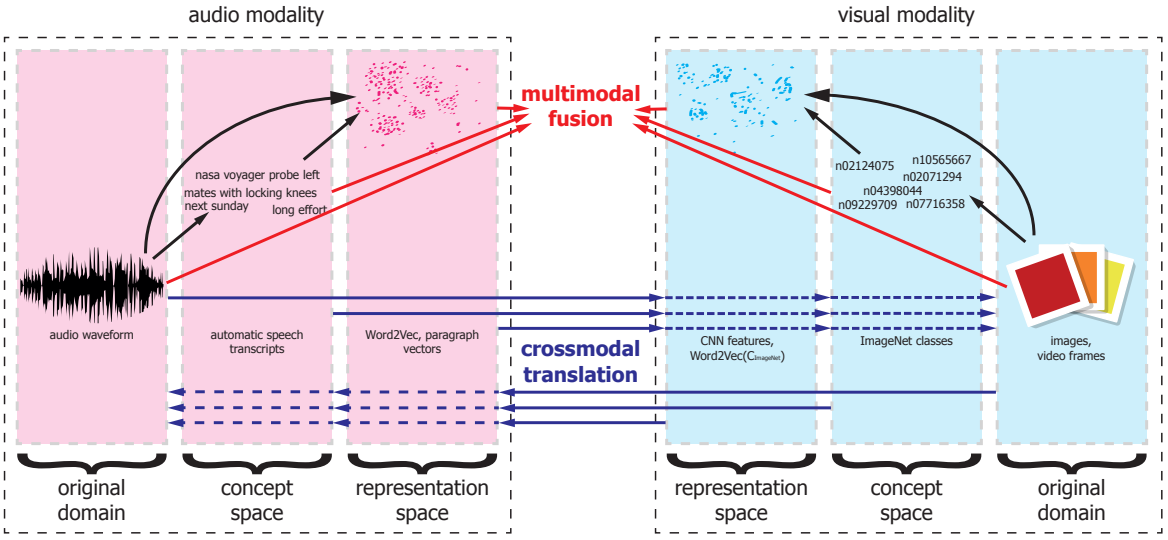


Figure 5.1: Two different input modalities (an audio modality and a visual modality), each presented at different presentation levels levels. Combining two modalities at arbitrary levels or multimodal fusion is denoted in red. Translating from one modality to another or crossmodal translation, at arbitrary levels, is denoted in blue.

are able to synthesize one modality from another, either at a representation level [30, 116] or even in the original domain [71, 90, 133, 117]. Most approaches that perform crossmodal translation also provide means to obtain multimodal fusion.

The two approaches are not necessarily disjoint. As we will show in Sections 5.2 and 5.4 crossmodal translation and multimodal fusion are tightly related and multimodal fusion can greatly benefit from methods focusing on crossmodal translation. Both crossmodal translation and multimodal fusion are not bound to modalities presented at the same level (e.g., images in a representation space and text in a representation space; see Figure 5.1) and can be performed with modalities at different levels. While there are many combinations, the most interesting ones use both modalities embedded into a representation space. We will use primarily the representation space in Sections 5.2 and 5.3. In Section 5.4, we will explore the possibility of using directly the original space for one modality as the original space is human-interpretable and can provide insight of the trained model.

5.1.1 Multimodal Approaches

Multimodal approaches create a joint representation of the initially disjoint modalities or otherwise merge the initial modalities without necessarily providing a bidirectional mapping of the initial representation spaces to the new representation space and back. These approaches are typically used in retrieval and classification tasks where translating back from the multimodal representation to the single-modal ones is not required.

A simple way to perform multimodal early fusion is by simply concatenating single-modal representations. This does not provide the best results, as each representation still belongs to its own representation space. It is also possible to utilize two separate modali-

ties by performing a linear combination [36] of the similarities obtained by comparing each of the two modalities. This late fusion avoids multimodal models and might offer slightly better results than simple concatenation. A linear combination can slightly correct the differences by giving more importance to one modality and implicitly reranking [100] similarity scores by different modalities. However, a linear combination requires cross-validation of the parameters, which often might be dependent on the specific dataset and the single modal representations used. In the next chapter, we use these two methods as a baseline to compare standard autoencoders and bidirectional deep neural networks against.

5.1.2 Crossmodal Approaches

Crossmodal approaches focus on bidirectional mapping of the initial representations [30], often by also creating a joint representation space in the process of doing so. They are able to map from one modality to another and back, as well as representing them in a joint representation space. These approaches can be used where crossmodal translation is required (e.g., multimodal query expansion, crossmodal retrieval) in addition to classification tasks.

5.2 Multimodal Autoencoders

Multimodal autoencoders are an extension of single-modal autoencoders that we previously introduced in Section 2.3 of Chapter 2. Two typical multimodal autoencoders are shown in Figure 5.2. The first (left) one illustrates a common approach that consists in concatenating the representations [73, 66] of the two modalities and training the autoencoder to reconstruct the data presented as input. The hidden layer in the middle is then used to obtain a joint multimodal representation (multimodal embedding). Except for the input (and thus the output) being a concatenation of the representations of two disjoint modalities, everything else is analogous to classical single-modal autoencoders.

The second (right) architecture is quite similar to the first one but consists of two separate inputs and outputs (one for each modality) and separate hidden layers in the initial and final layers. This is sometimes referred to as branch and we can say that this type of multimodal autoencoders have two separate branches, one for each modality. As in all the other autoencoders, one hidden layer in common is used for creating a joint multimodal representation. Sometimes, one modality is sporadically removed from the input to make the autoencoder learn to represent both modalities from one. The activations of the hidden layer are used as a multimodal joint representation. This enables autoencoders to also provide crossmodal mapping [73] in addition to a joint representation.

Autoencoders however have some downsides which slightly deteriorate performance:

- Both modalities influence the same central layer(s), either directly or indirectly, through other modality-specific fully-connected layers. Even when translating from one modality to the other, the input modality is either mixed with the other or with a zeroed input.
- Autoencoders need to learn to reconstruct the same output both when one modality is marked missing (e.g., zeroed) and when both modalities are presented as input.

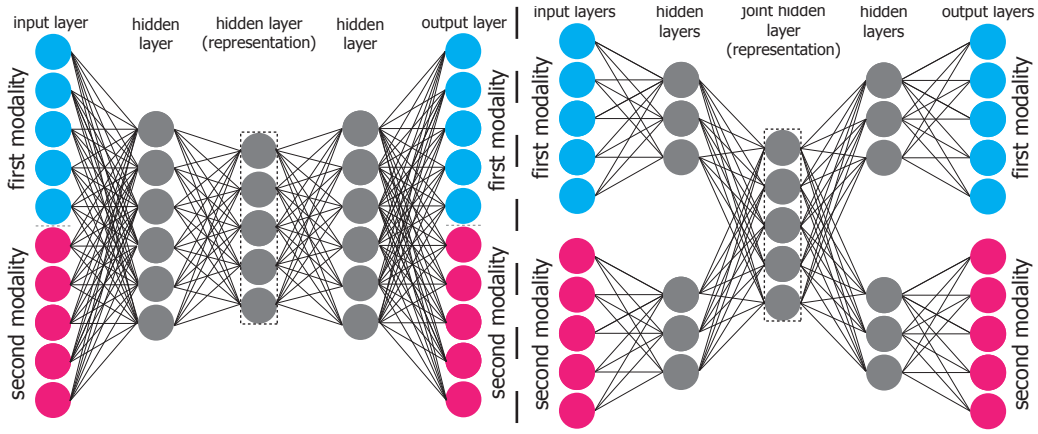


Figure 5.2: Two typical autoencoder architectures: left - concatenated representations at input and output, all hidden layers are joint; right - separated inputs, outputs and hidden layers, one hidden layer in common

- Classical autoencoders are primarily made for multimodal embedding while cross-modal translation is offered as a secondary function.

To address these issues, we propose bidirectional (symmetrical) deep neural networks [116], which we discuss next.

5.3 Bidirectional Deep Neural Networks

In bidirectional deep neural networks, learning is performed in both directions: one modality is presented as an input and the other as the expected output while at the same time the second one is presented as input and the first one as expected output. This is equivalent to using two separate deep neural networks and tying them (sharing specific weight variables) to make them symmetrical, as illustrated in Figure 5.3. Implementation-wise the variables representing the weights are shared across the two networks and are in fact the same variables. Learning of the two crossmodal mappings is then performed simultaneously and they are forced to be as close as possible to each other's inverses by the symmetric architecture in the middle. A joint representation in the middle of the two crossmodal mappings is also formed while learning and used to perform multimodal fusion. Symmetry is enforced solely in the central part given that a fully symmetric architecture would lose the flexibility to adapt to imperfect data and would converge very slowly and to a less optimal solution.

Formally, let $\mathbf{h_i^{(j)}}$ denote (the activation of) the hidden layer at depth j in network i ($i = 1, 2$, one for each modality), $\mathbf{x_i}$ the feature vector for modality i and $\mathbf{y_i}$ the output of the network for modality i. Networks are defined by their weight matrices $\mathbf{W_i^{(j)}}$ and bias vectors $\mathbf{b_i^{(j)}}$, for each layer j, and admit f as activation function. The entire architecture is then defined by:

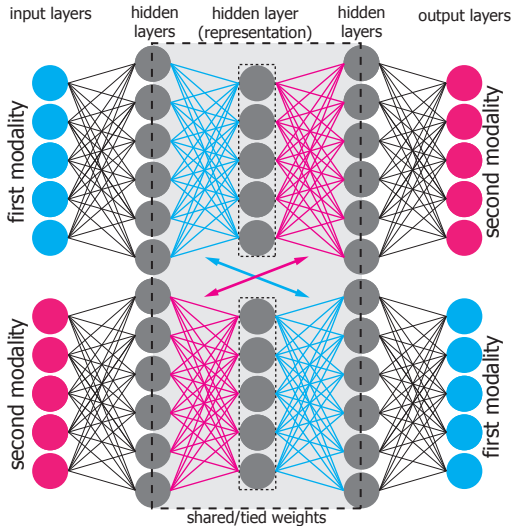


Figure 5.3: Proposed architecture: training is done in both directions; a shared representation is created by tying the weights (sharing the variables) and enforcing symmetry in the central part

$$\mathbf{h}_i^{(1)} = f(\mathbf{W}_i^{(1)} \times \mathbf{x}_i + \mathbf{b}_i^{(1)}) \qquad i = 1,2 \tag{5.1}$$

$$\mathbf{h}_1^{(2)} = f(\mathbf{W}^{(2)} \times \mathbf{h}_1^{(1)} + \mathbf{b}_1^{(2)}) \tag{5.2}$$

$$\mathbf{h}_1^{(3)} = f(\mathbf{W}^{(3)} \times \mathbf{h}_1^{(2)} + \mathbf{b}_1^{(3)}) \tag{5.3}$$

$$\mathbf{h}_2^{(2)} = f(\mathbf{W}^{(3)^T} \times \mathbf{h}_2^{(1)} + \mathbf{b}_2^{(2)}) \tag{5.4}$$

$$\mathbf{h}_2^{(3)} = f(\mathbf{W}^{(2)^T} \times \mathbf{h}_2^{(2)} + \mathbf{b}_2^{(3)}) \tag{5.5}$$

$$\mathbf{o}_i = f(\mathbf{W}_i^{(4)} \times \mathbf{h}_i^{(3)} + \mathbf{b}_i^{(4)}) \qquad i = 1,2 \tag{5.6}$$

It is important to note that the weight matrices $\mathbf{W}^{(2)}$ and $\mathbf{W}^{(3)}$ are used twice due to weight tying, respectively in Equations 5.2, 5.5 and Equations 5.3, 5.4. Training is performed by applying batch gradient descent to minimize the mean squared error of $(\mathbf{o_1}, \mathbf{x_2})$ and $(\mathbf{o_2}, \mathbf{x_1})$ thus effectively minimizing the reconstruction error in both directions and creating a joint representation in the middle.

Given such an architecture, crossmodal translation is done straightforwardly by presenting the first modality as $\mathbf{x_i}$ and obtaining the output in the representation space of the second modality as $\mathbf{o_i}$. A multimodal embedding is obtained by presenting one or both modalities ($\mathbf{x_1}$ and/or $\mathbf{x_2}$) at their respective inputs and reading the central hidden layers $\mathbf{h}_1^{(2)}$ and/or $\mathbf{h}_1^{(2)}$.

Multimodal embeddings are obtained in the following manner:

- When the two modalities are available, both are presented at their respective inputs and the activations are propagated through the network. The multimodal embedding is then obtained by concatenating the outputs of the middle layer.

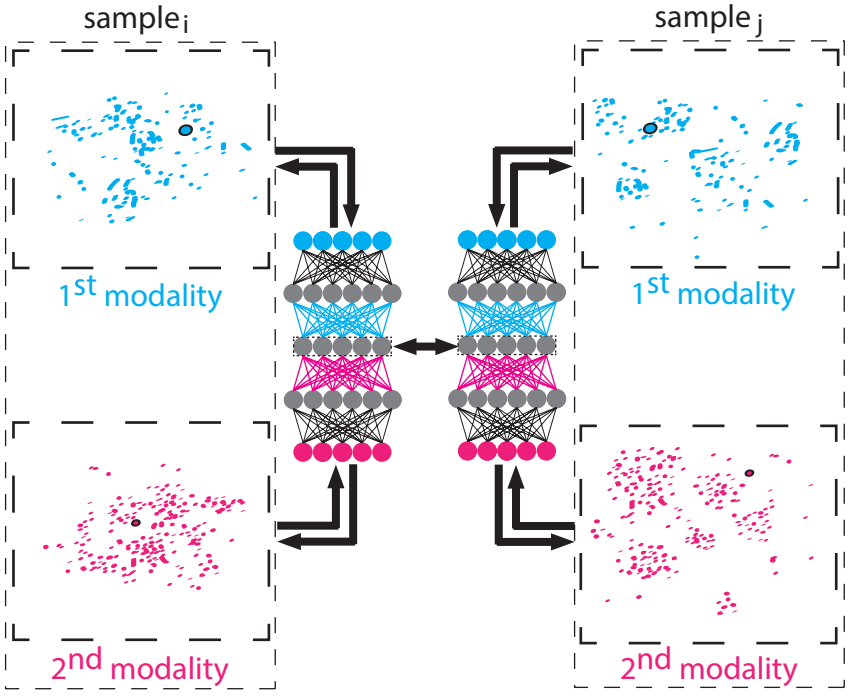


Figure 5.4: Computing the similarity of two data samples with BiDNNs: first a model is trained by learning crossmodal translations on every available sample in an unsupervised manner. After a trained model is obtained, for each of the two samples, each modality is presented and propagated through the networks. The activations in the central layers are than used to represent each sample with now fused modalities. Finally, the obtained multimodal embeddings are simply compared with a cosine distance to obtain their similarity.

- When one modality is available and the other is not, the available modality is presented to its respective input of the network and the activations are propagated. The central layer is then used to generate an embedding by being duplicated, thus still generating an embedding of the same size while allowing to transparently compare samples regardless of modality availability (either with only one or both modalities).

Note that while the embedding is multimodal, it corresponds to a space dedicated to cross-modal matching and thus significantly differs from classical joint multimodal spaces. Figure 5.4 illustrates the process of comparing the similarity of two samples (i and j) with bidirectional deep neural networks: for each sample, crossmodal translations between the two modalities are learned. Then, for the specific samples that we want to compare, their respective two modalities are presented at the inputs of a trained BiDNN model and their multimodal embeddings in the new, common representation space are formed. In the end, obtaining a similarity score of the two samples consists of simply computing the cosine distance of the newly obtained vectors.

5.4 Generative Adversarial Networks

In Sections 5.2 and 5.3 we performed crossmodal translation and multimodal fusion with continuous representations as inputs. Continuous representation spaces are well tailored to deep learning architectures and offer good characteristics [70] but are hardly human interpretable. The most human interpretable data presentation level is the original domain (e.g., the image space or the spatial domain). In this section, we propose the use of generative adversarial networks to perform crossmodal translations and multimodal fusion from a representation space to the original domain. More precisely, we evaluate text to image generative adversarial networks.

In all generality, the aim of a generative adversarial network (GAN) [35] is to estimate a generator that maps from a latent space to a particular data distribution. Such a data distribution of interest can be anything from a simple one dimensional statistical distribution [35] to a distribution of pixels in the spatial domain in natural images, conditioned on the surrounding pixels [131]. Architecture wise, generative adversarial networks belong to the set of architectures used in unsupervised learning (just like autoencoders discussed in Section 5.2 and bidirectional neural networks discussed in Section 5.3) and consist of a system of two networks:

- A **generative network** - $\mathbf{x}_g = \mathbf{G}(z, \Theta_g)$ that takes random noise z as input (to introduce stochasticity) and generates synthetic data samples $\mathbf{x}_g$ that mimic as closely as possible the distribution of the real data samples $\mathbf{x}$
- A **discriminative network** - $D_{out} = \mathbf{D}(x_{in}, \Theta_d)$ that takes both real data samples $\mathbf{x}$ and synthetic data samples $\mathbf{x}_g$ provided by the generative network and predicts whether the input is synthetic or real.

A generative adversarial network is trained by presenting the discriminator with real and synthetic data and updating its parameters Θ_d. When the discriminator is presented with a synthetic data from the generator, the output layer of the generator is the same as the input layer of the discriminator. In this situation, both the generative and the discriminative network can be seen as one network, and backpropagation can propagate further from the discriminative part to the generative part and update the parameters of the generative network Θ_g.

Generative adversarial networks can have the input of their generative network conditioned on a specific input variable. Such architectures are known under the name of conditional generative adversarial network or CGAN for short [71]. The conditioning input is simply concatenated to the noise input of the generative network. In this work we focus more on text to image mapping, as it is convenient for the task of video hyperlinking so we will focus more on generative adversarial networks that are suitable for this task. Figure 5.5 illustrates a generic text to image (conditional) generative adversarial network. The discriminative network (on the right) takes an image-text pair as input that can either be a real image with a continuous representation of its respective text or a synthetic image, generated by the generative network, with a textual continuous representation that is the same as the one provided to the generative network as input. The discriminative network is trained to differentiate between the two pairs. It is also possible to present a third pair to the discrim-

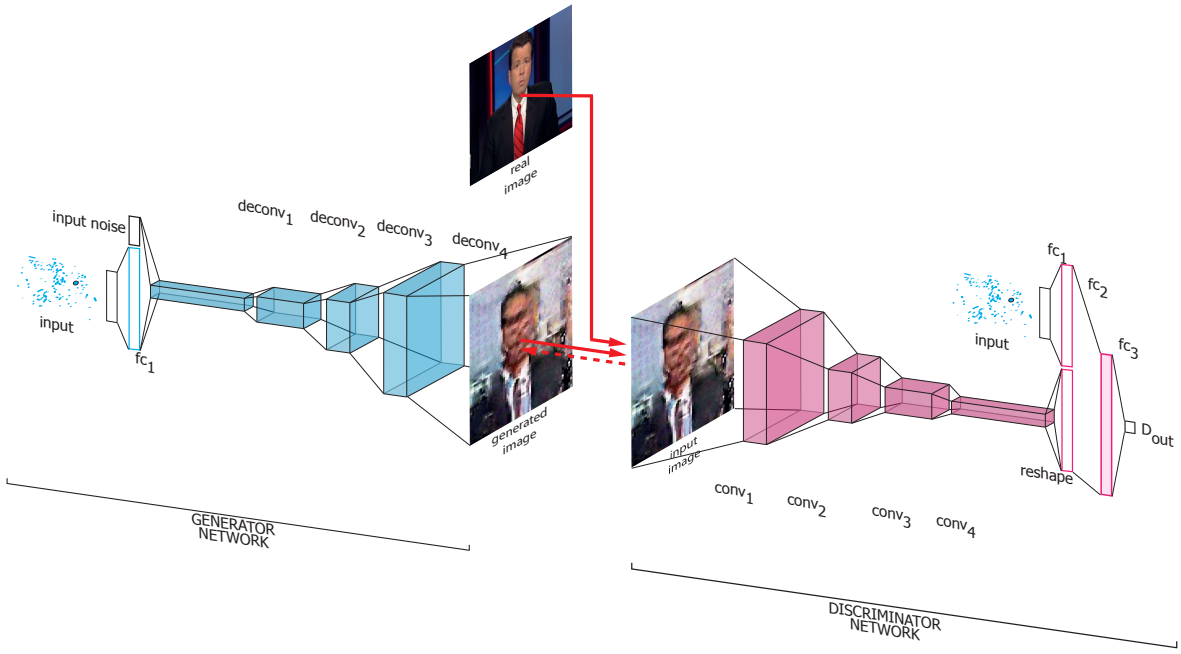


Figure 5.5: Text to image generative adversarial network: the generative network (on the left) takes a continuous representation of a text sample as input, together with a random noise input. The discriminative network (on the right), takes either a real image or an image from the generator and a continuous representation of its respective text sample and predicts whether the input image-text pair was a real or a synthetic one.

inator: a real image and an continuous representation of unrelated text. This is solely done to improve training and will be discussed in Section 6.7.

Although generative adversarial networks have been first introduced by [35] as a two-network, generative-discriminative model for generating high-quality, single-modal, realistic samples that could be mistaken for real samples from the dataset the model is trained on, they quickly became popular and are now used in a multitude of tasks, such as generating super-resolution images [26, 62], inpainting [131, 77], de-occlusion [134] and many others. Conditional GANs [71] have been shown to generate realistic samples of one modality, given a conditioning input of another modality. A typical example of a crossmodal CGAN model is text to image synthesis [90, 133] but the conditioned input is not necessarily bound to multiple modalities and can be conditioned also on the same modality [74]. In this work we, however, focus on the multimodal properties of conditional generative adversarial networks.

For the purpose of multimodal retrieval, we focus on crossmodal / multimodal conditional generative adversarial networks and ways to obtain meaningful multimodal representations from them. While multimodal setups are currently less explored, there is a lot of evidence that GANs learn meaningful representations in single-modal setups [67, 43, 16, 79], most notably in the generative network. These representations are obtained in a completely unsupervised manner and can be used to model changes in style, pose [16], color [79] or even style and structure in RGBD data [124]. Evidence suggests that both the generative network and the discriminative network can produce meaningful single-modal representations [85].

Our work falls into the category of text to image CGANs [90, 133], where we explore the possibility of obtaining good multimodal representations from the discriminator, while using the generator to visualize crossmodal mappings for the purposes of multimodal retrieval in video hyperlinking and to improve the embeddings from the discriminator network.

The CGAN architecture we propose is based on the GAN-CLS text to image model [90]. The model consists of a generative network G and a discriminative network D. The generative network takes a noise vector z, sampled from a $\mathcal{U}(-1, 1)$ distribution and a text embedding φ, and generates a synthetic image $\hat{x} = G(z, \varphi)$. The generator consists of a separate fully-connected layer with a leaky-ReLU activation function (that allows for a small non-zero gradient on the negative side) for modeling the text embeddings. This is then concatenated with the noise input. A standard deconvolutional network follows. The discriminator network takes an image x and a text embedding φ and determines whether the pair is real or artificial $D(x, \varphi)$. Contrary to the basic conditional generative adversarial network model that only differentiates between real and synthetic images, CGAN-CLS is trained on three pairs. In our case {real image, real text}, {incorrect image, real text} and {synthetic image, real text}. As we are doing unsupervised learning, pairs with a non-matching image and text are chosen at random and the dataset is not split. Pairs of real matching and non-matching text-image pairs are necessary to train the discriminator. The discriminator consists of a series of convolutional layers with batch normalization, followed by a leaky-ReLU for modeling the input image and a full-connected leaky-ReLU layer to model the text embeddings. The two branches are then concatenated and a 1×1 convolution is performed, followed by batch normalization and a leaky-ReLU activation before obtaining the discriminator score. For multimodal embedding, we use the vector obtained after the final 1×1 convolution of the discriminator. To maintain the losses of both the generator and the discriminator at similar levels (having a discriminator that is performing too well would prevent the generator from converging), the generator is updated four times more often than the discriminator. A cosine distance between the obtained multimodal embeddings is used to measure the similarity of the desired video segments both in the case of multimodal autoencoders and conditional generative adversarial networks.

5.5 Conclusion

In this chapter, we described methods to perform crossmodal translation and multimodal fusion on a theoretical level. We first described classical multimodal autoencoders in Section 5.2 and then we described our modified architecture, bidirectional deep neural networks, where multimodal fusion is improved by focusing on crossmodal translations in Section 5.3. Both methods work with both modalities presented as continuous representations at their inputs. In order to synthesize one modality directly into the original domain (the spatial domain or image domain), we discussed the possibility of using conditional generative adversarial networks to perform both multimodal fusion and crossmodal translations, also on a theoretical level. In Chapter 6, we will introduce the task of video hyperlinking, its respective dataset, evaluate the methods empirically and analyze their performance on a real life scenario.

Chapter 6

Video Hyperlinking

Contents

In Chapter 5 we introduced classical multimodal approaches, such as multimodal autoencoders, and proposed our methods, bidirectional deep neural networks and a method based on generative adversarial networks, from a theoretical standpoint. Given the interest of our team in video content related tasks and a history of participating at international benchmarks that evaluate methods in the task of video hyperlinking, we decided to evaluate our previously proposed methods in the task of video hyperlinking and to participate to an international benchmarking initiative that focuses on video hyperlinking.

In this chapter, we first introduce the task of video hyperlinking in Section 6.1 and its related datasets in Section 6.2. In Section 6.3 we introduce the metrics used to evaluate relatedness of video segments in the task of video hyperlinking. This metric will then be used in Sections 6.4 to 6.7 where we will evaluate different approaches to video hyperlinking.

6.1 The Video Hyperlinking Task

With the increasing abundance of professional and community-based video content, it is important to not only offer ways to discover videos through search queries but also to allow

Figure 6.1: The task of video hyperlinking: the anchor represents a video segment that a user is currently viewing and wishes to find related video segments (targets). In this example, an anchor where a fish & chips restaurant is displayed is linked to a target containing a part of a cooking show where a recipe for preparing fish & chips is illustrated thus linking the two video segments through different modalities.

for an explorative approach that proposes to a user a set of potentially interesting video segments, given the video segment that is currently viewed. This arises the need for video segments to be hyperlinked within a multimedia data collection. The seminal idea of video hyperlinking is to create hyperlinks between different videos and/or video segments based on their content.

In task of video hyperlinking, there are two main concepts: anchors and targets. Anchors represent segments of interest within videos that a user would like to know more about. Targets represent potential segments of interest that might or might not be related with a specific anchor. The goal is to hyperlink relevant targets for each anchor by using multimodal approaches. An example of the task of video hyperlinking is illustrated in Figure 6.1 where an anchor is linked to a target though the content linking them is present in different modalities. The anchor is a video segment, part of a video about English culture, where a fish & chips restaurant is displayed and is thus containing the topic of interest in the visual modality. The target is a video segment, part of a cooking show, where the host is explaining how to prepare a fish & chips dish without having many visual clues, until the end and containing most of the clues for the topic of interest in the speech modality.

Each video consists of at least two data streams: a visual stream and an audio stream. A visual stream is represented by a set of consecutive images (frames) of which the most meaningful ones are keyframes. Keyframes (also known as intra-frames) are fully stored frames - frames where the complete information is stored in the video stream. Other frames (known as inter-frames) are expressed as a change from neighboring keyframes. This is due to the fact that, in most videos, neighboring frames contain a lot of redundant information. Keyframes provide the whole frame in the beginning, after the accumulated changes from the original previous keyframe are too big and after every scene change. These properties make keyframes a good source of visual information from where visual concept extraction, visual embedding, action or event recognition and other visual content analysis methods can be performed. Audio streams also provide information - most often, but not limited to, as speech. After automatic transcription, the audio part of a video sequence is typically used and further processed as a sequence of words. Data from an audio source does not have to correlate with data from the corresponding video source but it certainly can. Given

this nature of videos and/or video segments, it is necessary to perform content analysis and comparison of both visual information and spoken information both in a crossmodal and in a multimodal fashion (e.g. a link between two video segments can reflect a connection between a concept being discussed in the first video segment and a location being displayed in the second video segment).

6.2 Datasets

We participated in a number of international benchmarks that evaluate methods in the task of video hyperlinking and used their datasets in both online (full video hyperlinking setup) and offline (problem downscaled to multimodal retrieval) evaluations. Given the evolution of the dataset and the video hyperlinking initiative over time, there are mainly two datasets that we refer to in this work: i) MediaEval's video hyperlinking dataset, more specifically the dataset and the groundtruth formed post-hoc after MediaEval 2014 and ii) TRECVID's video hyperlinking dataset from 2016, used in the live evaluations of the proposed systems. The video hyperlinking challenge was originally part of the Medieval initiative [29]. In 2016, the video hyperlinking challenge moved to TRECVID [3]. Additionally, while the dataset originally used videos segments provided by BBC, since 2016 the dataset is formed from video segments provided by BlipTV.

In this work, we use the MediaEval 2014 dataset to evaluate different single-modal representations, multimodal autoencoders, bidirectional neural networks and generative adversarial networks with a fixed groundtruth that comes with the dataset. The TREVID 2016 dataset was used for the live evaluation of bidirectional neural networks through the challenge, thus lifting the restriction of having a predefined groundtruth.

6.2.1 MediaEval 2014

The original data consists of approximately 2,700 hours of BBC broadcasted content. For each video, multiple data and modalities are available. As described in Section 6.1, video segments consist of anchors and targets. In practice, targets are not given and have to be defined automatically before assessing their relevance to each of the 30 anchors provided. Evaluation of relevance is thus done post-hoc on Amazon Mechanical Turk (AMT). After the completion of the benchmark, an annotation with the given anchors and all the targets proposed by the participants, as well as their relevance for the given anchor is provided. This annotation is officially provided after the evaluation and is very useful to evaluate different methods by using it as a groundtruth in a multimodal retrieval setup. In this case, the task of video hyperlinking is downscaled by removing the task of creating video segments and proposing relevant one, and is evaluated through multimodal retrieval with a given groundtruth.

Multimodal retrieval evaluation task thus consists of using multimodal information to rank the targets by relevance for each anchor and comparing their relevance with the previously established groundtruth. In total, the dataset consists of 30 anchors, 10,809 targets and a ground truth with 12,340 anchor-target pairs (either related or unrelated). Interestingly, among the anchor and target segments, not all have both transcripts and visual concepts

available. Regarding keyframes, there are in total 371,664 keyframes for an average of 34.3 keyframes per video segment.

We used a combination of two modalities: either automatic transcripts of the audio track and KU Leuven [110] visual concepts or automatic transcripts of the audio track and descriptors of each keyframe obtained with different convolutional neural network architectures. Both automatic transcripts and KU Leuven visual concepts are provided as part of the dataset. KU Leuven visual concepts consists of multiple ImageNet [92] classes detected in each keyframe with a CNN architecture and provided as a textual description together with each keyframe.

6.2.2 TRECVID 2016

The dataset under scrutiny at TRECVID 2016 is the BlipTV dataset [3], composed of 14,838 videos with a duration of 13 minutes on average. These videos span multiple languages including English, Chinese, Arabic, etc.

We considered all languages while training our models, while only English anchors were selected by the organizers. It can be expected that considering multiple languages lowered the live performance of our evaluated systems described in the following sections. We used the automatic transcriptions provided by LIMSI [33]. The dataset also provides metadata and shot boundaries that we do not use. We chose not to use the user-generated metadata with the objective to be as close as possible to a fully automatic system that could be transposed to any video dataset without such metadata. We exploit a speech-based rather than a shot-based segmentation, in order not to cut a segment in the middle of a sentence. The segmentation was performed by taking only 30 seconds of contiguous speech and then cut at the following speech pause as detected by the speech transcription system. We run this speech-based segmentation process twice, using an offset of one speech segment at the second pass, in order to obtain an overlapping segmentation. This resulted in 307,403 video segments with a mean duration of 45 seconds.

6.3 Assessing Relatedness

The main evaluation metric in video hyperlinking is the relatedness of the proposed targets to each queried anchor. For the whole set of anchors (the queries to a retrieval system), the proposed targets are evaluated either relevant or non relevant and the precision of the system is calculated. Additional restrictions are imposed to prevent the systems from proposing targets that belong to the same video as the anchor and to prevent the systems from proposing overlapping targets and thus increasing their precision without actually retrieving different videos. Typically, in the previously mentioned benchmarking initiatives, only the top N proposed targets are evaluated and this restriction is then also implicated in offline evaluations by the dataset, in a multimodal retrieval setup. This fits a realistic scenario where the user would only be interested in a few proposed video segments and would not search a large collection to find videos related to the currently viewing video. In all video hyperlinking challenges (and thus also in the post-hoc groundtruth and the official evaluation tool) prior to 2016, the top-10 proposed targets were evaluated and the precision at 10 for each system

was evaluated. In TRECVID's 2016 video hyperlinking challenge, the organizers opted to evaluate solely the top-5 proposed targets and report precision at 5 for each system.

6.4 Single-Modal Approaches to Video Hyperlinking

In this section, we analyze methods for obtaining good single modal representations. We start with different methods to represent automatic audio transcriptions of the video segments and we progress to methods to represent keyframes of the video segments. In the following sections we will evaluate different methods to create joint multimodal representations, as well as allowing for crossmodal translation. Where appropriate, for some single-modal cases, methods for aggregating multiple embeddings into one single-modal representation are also tackled.

6.4.1 Initial Single-modal Representations

All methods presented in this chapter utilize two data modalities: i) automatic audio transcripts and ii) video keyframes. Automatic audio transcripts are used instead of subtitles which are not always available in practice and would include a human component in the system. Video keyframes are considered in two different settings: using *ImageNet* concepts [92] or directly describing images with features obtained with state-of-the-art convolutional neural networks.

Automatic transcripts of a video segment consist of one or more sentences, each with multiple words. This makes sentence/paragraph/document representation methods suitable for the task. Two methods were evaluated (each in different settings): paragraph vectors [59] and Word2Vec [70]. Contrary to paragraph vectors, Word2Vec is not specifically designed for embedding bigger blocks of text. However it was shown that Word2Vec can perform quite well [13] and can be suitable when combined with an aggregation of the embedded words. During training, visual concepts can either be sorted, duplicated and shuffled multiple times or left in the order of probabilities of their presence in the image. All methods were trained directly on the automatic transcripts, in an unsupervised manner.

For each keyframe of each video segment, a set of top scoring visual concepts is used as information indicating what's visible in the image. Visual concepts describe a class of objects

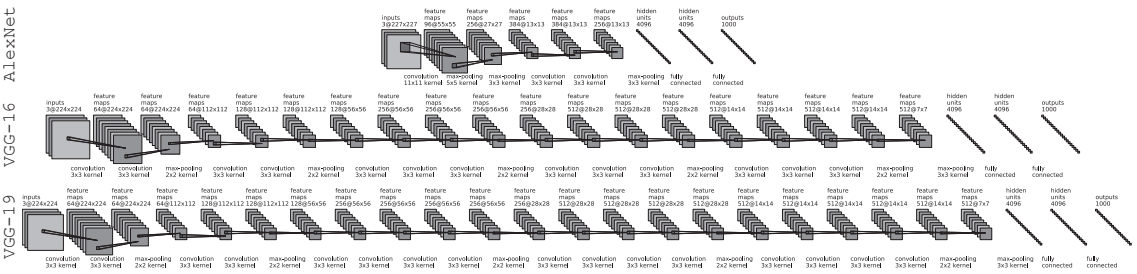


Figure 6.2: Simplified comparison of the CNN architectures used in this work: AlexNet (top), VGG-16 (middle) and VGG-19 (bottom). For simplicity, only the main layers are shown. Merging, reshaping, padding and other layers are not illustrated.

or entities: e.g., *"n02121808"* indicates *"Any domesticated member of the genus Felis (Domestic cat, house cat, Felis domesticus, Felis catus)"* and includes all related subcategories. We treat each visual concept as a word and utilize it to obtain word embeddings (with Word2Vec or paragraph vectors) representing the visual information of a video segment provided by its visual concepts in a continous representation space.

While visual concepts are typically given with the datasets provided by video hyperlinking benchmarking initiatives, they are not optimal for deep learning methods as they are bag-of-words representations and not continuous representations. Convolutional Neural Networks (CNNs) provide state-of-the art visual descriptors [98] that have been shown to perform well both in computer vision applications [51, 130] and in video summarization tasks [45]. In this work, we test three different state-of-the art deep convolutional neural network architectures, namely AlexNet, VGG-16 and VGG-19. Figure 6.2 illustrates, in a simplified manner (only main layers are shown: convolutional, pooling and fully-connected layers), the architectures of such networks. AlexNet [51] is deep convolutional neural network of medium depth, with 3 convolutional layers, 3 max-pooling layers and a set of fully-connected layers at the end. VGG networks [101] are very deep convolutional neural network architectures defined by the *Visual Geometry Group*. We use two VGG architectures, namely VGG-16 and VGG-19, with 16 and 19 "weight layers" respectively. The VGG-16 architecture consists of 13 convolutional layers, 5 max-pooling layers and 3 fully-connected layers. The VGG-19 architecture consists of 16 convolutional layers, 3 max-pooling layers and 3 fully-connected layers.

Depending on the subtask and the method used, the resulting representations might require aggregation, e.g., to represent all the automatic transcripts of a video segment with Word2Vec or to represent all the keyframes of a video segment. Some methods, on the other side, do not require additional aggregation (e.g., paragraph vectors). In this work, we tested two means of aggregating descriptors: simple averaging [13] and Fisher vectors [80, 81].

6.4.2 Video Hyperlinking with the Original Representations

We chose to represent the transcripts and visual concepts of each anchor and target with a *Word2Vec* skip-gram model with hierarchical sampling [70], a representation size of 100 and a window size of 5. The visual concepts were sorted previous to learning and the representations of the words and concepts found within a segment were averaged [13]. This option worked best for our task.

Convolutional neural network representations were obtained by using the output of the last fully-connected layers of AlexNet, VGG-16 and VGG-19, respectively. All three convolutional neural network architectures yield a representation of size 4096. Since there are multiple keyframes in each video segment, aggregation was either done by averaging or by using Fisher vectors. The average proved to provide solid representations based on AlexNet, as well as the best representations, based on VGG-16 and VGG-19. For AlexNet, Fisher vectors provided slightly better results (with a previous dimensionality reduction with PCA to a size of 64 and GMM with 64 mixtures). Averaged VGG-16 provide the best visual embedding, yielding a result of 70.67 % in precision at 10. A standard cosine distance is used in all the experiments as a measure of similarity. The performance of the different methods is shown in Table 6.1.

Given that we have two modalities at our disposal, it seems reasonable to use both to

Representation	Aggregation	P@10 (%)
Automatic transcripts		
Word2Vec	**average**	**58.67**
Word2Vec	Fisher	54.00
PV-DM	-	45.00
PV-DBOW	-	41.67
Visual information		
KU Leuven visual concepts, Word2Vec	**average**	**50.00**
KU Leuven visual concepts, PV-DM	-	45.33
KU Leuven visual concepts, PV-DBOW	-	48.33
AlexNet	average	63.00
AlexNet	**Fisher**	**65.00**
VGG-16	**average**	**70.67**
VGG-16	Fisher	64.67
VGG-19	**average**	**68.67**
VGG-19	Fisher	66.00

Table 6.1: Single modal representations of automatic transcripts and visual information

determine the similarity of two video segments. There are simple ways to combine two modalities without appealing to multimodal fusion. A simple way to combine two modalities is by simply concatenating single-modal representations. This does not provide the best results, as each representation still belongs to its own representation space. It is also possible to utilize two separate modalities by performing a linear combination [36] of the similarities obtained by comparing each of the two modalities (sometimes called score fusion). This late fusion avoids multimodal models and might offer slightly better results than simple concatenation (a linear combination can slightly correct the differences by giving more importance to one modality and implicitly reranking [100] similarity scores by different modalities). However, a linear combination requires cross-validation of the parameters, which often might be dependent on the specific dataset and the single modal representations used. We use these two methods as a baseline to compare standard autoencoders and bidirectional deep neural networks against.

Table 6.2 illustrates the performance of the previously mentioned methods utilizing two modalities without performing multimodal fusion. There is no significant improvement when concatenating embedded transcripts and visual concepts. However, a simple concatenation of embedded transcripts and embeddings obtained with convolutional neural networks improves over each single-modal representation alone. For instance, combining VGG-16 embeddings with embedded transcripts yields 75.33 % (precision at 10) over the initial performance of 70.67 % and 58.67 % respectively. A linear combination of similarities, on the other hand, does not offer a multimodal embedding but might be simpler (often used for relevance reranking) over simple concatenation, at the cost of having to optimize the parameters on another dataset and possibly at the cost of higher variance.

Modalities	Method	P@10 (%)	σ (%)
Simple multimodal approaches			
Transcripts, visual concepts (word2vec)	concatenation	58.00	-
Transcripts, AlexNet	concatenation	70.00	-
Transcripts, VGG-16	concatenation	75.33	-
Transcripts, VGG-19	concatenation	74.33	-
Transcripts, visual concepts (word2vec)	linear combination	61.32	3.10
Transcripts, AlexNet	linear combination	67.38	2.66
Transcripts, VGG-16	linear combination	71.86	4.11
Transcripts, VGG-19	linear combination	71.78	3.90

Table 6.2: Comparison of simple methods of utilizing two modalities in terms of precision at 10 (P@10). When linearly combining the scores, we run the experiment for many possible linear scores (without cross-validation on a separate dataset) and report the average and the standard deviation.

6.5 Video Hyperlinking with Multimodal Fusion

Multimodal autoencoders are the most common current method for obtaining multimodal embeddings. We implemented the model with separate branches for each modality, as previously described in Section 5.2 of Chapter 5. We implemented the described autoencoder with a central layer of size 1000. Bigger sizes did not improve the results but smaller ones did deteriorate them. The inputs, outputs and their associated fully-connected layers were sized accordingly with the dimensionality of the input data.

Table 6.3 reports the results. It can be clearly seen that multimodal embedding performs better than each single modality by itself; e.g. combining embedded transcripts and VGG-19 features yields 74.73 %, compared to 58.67 % and 68.07 % respectively. However, in some cases, it seems that embeddings obtained in such a way do not yield significantly better results than simple methods. We believe this is caused by the already good single-modal representations and the fact that autoencoders have to train to represent the correct output with both modalities being present at their input and with one zeroed modality. In cases where the initial embeddings perform less (e.g., embedded visual concepts combined with embedded transcripts), autoencoders seem to improve in a more significant way.

Modalities	Method	P@10 (%)	σ (%)
Multimodal autoencoders			
Transcripts, visual concepts		59.60	0.65
Transcripts, AlexNet		69.87	1.64
Transcripts, VGG-16		74.53	1.52
Transcripts, VGG-19		75.73	1.79

Table 6.3: Performance of multimodal autoencoders given different representations for each modality. For each case, precision at 10 (%) and the standard deviation are reported.

6.6 Multimodal Fusion Through Crossmodal Translations

Bidirectional deep neural networks are used in video hyperlinking as illustrated in Figure 6.3: first, for all video segments containing two modalities, both are taken (embedded automatic transcripts with either embedded visual concepts or embedded CNN representations) and crossmodal translations between the two modalities are learned. To compare the similarity of two video segments through multimodal fusion, their respective two modalities are presented at the inputs of a trained BiDNN model and their multimodal embeddings in the new, common representation space are formed. The two multimodal embeddings are then simply compared with a cosine distance to obtain a similarity measure.

6.6.1 BiDNN Multimodal Embedding

We implemented a bidirectional deep neural network comparable with the previously described autoencoder: a central fully-connected layer yielding a representation of size 1000 and inputs/outputs dependent on the modalities used. Bidirectional deep neural networks behaved similarly to autoencoders as representation sizes bigger than 1000 did not bring any significant improvement. Smaller ones had deteriorated performance. This confirms the choice of the dimensionality of the new multimodal representation by two independent methods. Each bidirectional deep neural network was trained with five independent runs of 1000 epochs each, although they converged earlier, the results were averaged and, together with their respective standard deviations, are reported in Table 6.4. Significance levels of improvements are computed with single-tailed t-tests and reported where appropriate.

This provides superior multimodal embeddings that bring significant improvement. For instance, combining embedded transcripts with VGG-19 embeddings yields a precision at 10 of 80.00 %, compared to 58.67 % and 68.67 % respectively. All the other tested combinations also yielded better results and high quality multimodal embeddings.

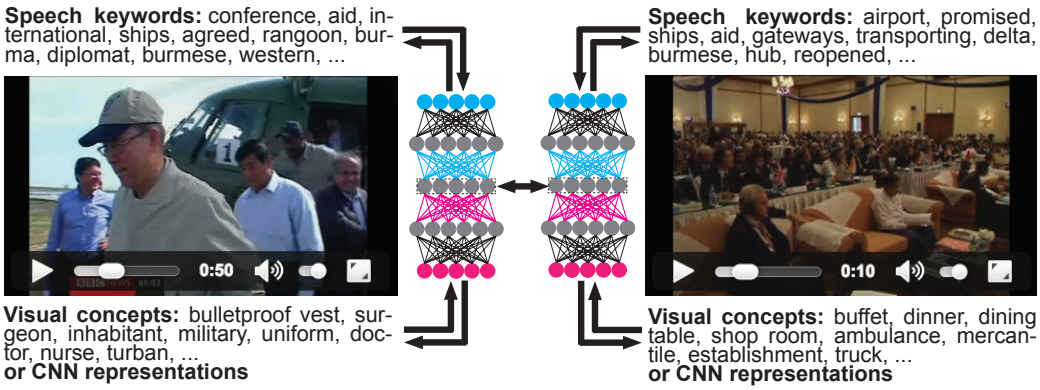


Figure 6.3: Video hyperlinking with bidirectional deep neural networks: two video segments, both with two modalities (automatic transcripts and either KU Leuven visual concepts or CNN features of each keyframe) are compared after their multimodal embeddings are computed

6.6.2 BiDNN Single Modality Embedding

Although bidirectional deep neural networks are trained in a multimodal setup, it is possible to embed only one modality by presenting it to the respective input and propagating the activations forwards until the central, representation layer. Doing so might offer an insight about the new representation space, common for both modalities, and its performance compared to the original representation spaces. Results, given in Table 6.4, clearly show that each newly formed common representation space is significantly better than its respective original representation space. Automatic transcripts improve from 58.67 % to 66.78 %, visual concepts from 50.00 % to 54.92 % and VGG-19 embeddings from 68.67 % to 70.81 %. These results are obviously not as good as multimodal embeddings obtained by combining two modalities but they clearly show the improvement that bidirectional deep neural networks bring even when used in a single-modal fashion and not only as a common space where representations from originally different representation spaces are projected. This is due to the fact that those new representations are not completely single-modal as they have been projected into a new space by a projection that has been learned in a crossmodal fashion.

6.6.3 BiDNN Crossmodal Query Expansion

Bidirectional deep neural networks naturally enable crossmodal expansion; meaning a missing modality is filled in by translating from the existing one. If a transcript is not available for a video segment, it is generated from the visual concepts and conversely. Using crossmodal query expansion so that all segments have all modalities, we obtain, 62.35 %, when combining transcripts and visual concepts. There are no significant improvement when using crossmodal query expansion with pairs that consist of representations obtained with high-performing deep convolutional neural networks. This is due to the relatively small number of samples with one missing modality, so filling the missing modalities does not have a big impact - especially with well performing single-modal representations. The original representation used perform less good, as shown in Sections 6.6.1 and 6.6.2, than the new common representation spaces obtained with bidirectional deep neural networks and the influence of filling-in the few video segments with missing modalities is almost insignificant.

6.6.4 Video Hyperlinking Evaluation of Bidirectional Neural Networks

In the previous subsections, all evaluations were performed with a static groundtruth. This means that all the evaluated systems were in fact limited to reranking the targets (both related and unrelated) in the groundtruth [119]. Targets were proposed by the participants of MediaEval's 2014 video hyperlinking challenge. To truly evaluate the performance of our proposed method, bidirectional deep neural networks, in the task of video hyperlinking, we participated at an international benchmarking initiative, now under TRECVID and with videos from BlipTV. By doing so, the system is able to propose targets from all the possible video segments (307,403 of them, given our speech based segmentation) as long as they don't belong to the same video as the anchor and two proposed targets do not overlap (if they do, the lower scoring one is removed and only the higher scoring one is proposed).

Regarding the initial, single-modal representations, we chose to represent the transcripts of each anchor and target with a *Word2Vec* skip-gram model with hierarchical sampling [70],

Modalities	Method	P@10 (%)	σ (%)
BiDNN single modality embedding			
Transcripts		66.78	1.05
Visual concepts		54.92	0.99
AlexNet		66.33	0.58
VGG-16		68.70	1.98
VGG-19		70.81	1.08
BiDNN multimodal embedding			
Transcripts, visual concepts		73.74	0.46
Transcripts, AlexNet		73.41	1.08
Transcripts, VGG-16		76.33	1.60
Transcripts, VGG-19		**80.00**	**0.80**
BiDNN query expansion			
Transcripts, visual concepts		62.35	0.25
Transcripts, AlexNet		70.11	1.25
Transcripts, VGG-16		75.33	0.10
Transcripts, VGG-19		74.33	0.10

Table 6.4: Performances of fusion (multimodal embedding), single modal embedding and crossmodal translation (query expansion) with bidirectional deep neural networks and different input representations for each of the two modalities. Precision at 10 (%) and the respective standard deviation is reported for each case.

a representation size of 100 and a window size of 5, as this has been shown to achieve the best performance in the previously mentioned offline (multimodal retrieval) evaluations. Visual embeddings are obtained from a very deep convolutional neural network (CNN) VGG-19 [101], pretrained on *ImageNet*. The last fully-connected layer is extracted and used to represent the visual information. Therefore, we obtain a 4096 dimensional embedding for each keyframe that is later averaged over all the keyframes to represent the whole video segments.

We submitted three runs in such a way to have a clear idea of how much does multimodal fusion with bidirectional deep neural networks improve over the initial modalities:

- a single-modal run using only embedded speech transcripts
- a single-modal run using only averaged VGG-19 features to represent each video segment
- a BiDNN-fused multimodal run combining the two representation into a new, fused one

For all the three runs, we just used cosine distances over the representations to determine the similarity of each video segment evaluated as a potential target for each given anchor. As previously mentioned in Section 6.2, at TRECVID 2016, precision at 5 is reported instead of the previously used precision at 10.

Method	Precision at 5 (%)
Our methods	
Only visual (averaged VGG-19 features)	45
Only speech transcripts (averaged Word2Vec)	40
BiDNN multimodal fusion (visual and speech)	**52**
All participants	
Maximum	**52**
Upper quartile	41
Median	35
Lower quartile	32
Minimum	24

Table 6.5: Results of the live, video hyperlinking evaluation at TRECVID 2016. Results of our three systems and the statistics of all participants.

Table 6.5 illustrates the performances of our three systems, as well as the statistics of all participants. The initial single-modal speech and visual representations obtained results of 40% and 45% respectively. Multimodal fusion performed in a crossmodal fashion through bidirectional deep neural networks managed to improve the results and obtain 52 % in terms of precision at 5. This was also the best performing method at TRECVID's 2016 video hyperlinking task and defines the new state of the art for the task.

6.7 Video Hyperlinking and the Original Domain

Bidirectional deep neural networks (BiDNNs) offer improved multimodal fusion by focusing on crossmodal translations first. They intrinsically model crossmodal translations from different modalities given as continuous representations. While it is possible to analyze a learned model and its crossmodal translations by feeding one modality to the input and finding samples that are close to the translated modality into the representation space of the other modality, this is inherently more difficult and less clear for a human observer than seeing directly the result in the original domain. In this section, we use conditional generative adversarial networks, previously described in Section 5.4, to learn a crossmodal translation that goes from a continuous representation space of speech transcripts and synthesizes directly into the original spatial domain (image domain). We also study the possibility of using such a model to perform multimodal fusion and expect an improvement due to the virtually unlimited new synthetic samples provided by the generator part of a generative adversarial network.

Generative adversarial networks are complex to train and are currently limited in regards of the image size they can synthesize. We will still use the dataset of MediaEval 2014, as in Sections 6.4 to 6.6 but now with image sizes of 64×64 pixels, while keeping everything else the same. To have a fair comparison, we will reevaluate multimodal autoencoders and bidirectional deep neural networks with the same image sizes of 64×64 pixels. The CGAN model works directly with the previously described *Word2Vec* text embeddings and images. Multimodal AEs also need visual embeddings as inputs. We obtained them, as before, from

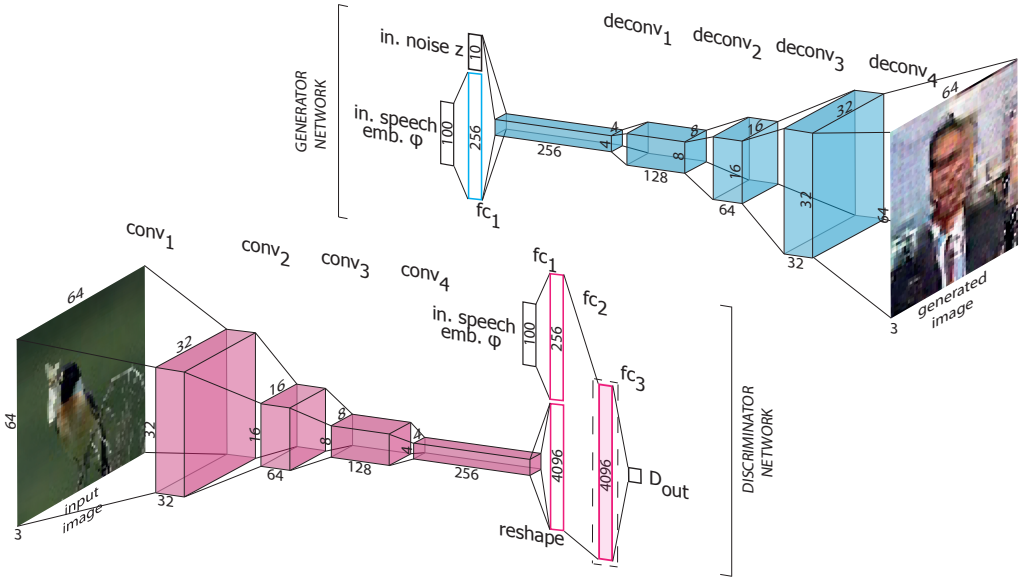


Figure 6.4: Used CGAN architecture, consisting of a generator and a discriminator network. Multimodal embeddings are obtained in the discriminator, after the last, 1D convolution operation, as denoted with a dashed rectangle.

the same images, with a VGG-19 network (pretrained on ImageNet) and they are of size 4096. Table 6.6 shows the initial single-modal results, which are 56.55% for speech transcripts and 52.41% for video keyframes, in terms of precision at 10 [3].

6.7.1 Multimodal Fusion with CGANs

To evaluate the feasibility of obtaining multimodal embeddings with CGANs, we used the architecture described in Sec. 5.4 with a speech embedding of dimension 100, a noise input of size 10, a fully-connected layer for modeling text of size 256 and 4 *"deconvolutional"* layers of increasing size, each with 256, 128, 64 and 32 feature maps respectively in the generator network. The output of the generator is an image of size 64×64. The discriminator network consists of an image input of size 64×64 followed by 4 convolutional layers with batch normalization, each with 32, 64, 128 and 256 feature maps respectively and decreasing sizes and a speech embedding input of size 100, followed by a fully-connected layer of size 256. The two branches are then concatenated and followed by another, final convolution. We trained the network for 1000 epochs by using the Adam optimizer with a learning rate r of 0.0001, a momentum β of 0.5 and a batch size of 64. We use the one-dimensional convolutional layer in the discriminator that follows the merging of the two branches, and proceeds the batch normalization and activation layers, as a multimodal embedding layer, as illustrated in Figure 6.4. Other layers did not perform well and provided a lower or equal quality than the initial single-modal inputs. It is also possible to train the discriminator by itself, solely on real and wrong image-speech pairs. This does not converge to a comparably good enough solution. The generator network thus does not only provide a mean to visualize crossmodal mappings but aids in obtaining high-quality embeddings in the discriminator by providing additional synthetic samples. The generator network does not only provide a mean to visu-

Representation	Precision at 10 (%)	σ (%)
Speech Transcripts Only	56.55	-
Visual Only (VGG-19)	52.41	-
Multimodal AE	57.94	0.82
BiDNN	59.66	0.84
CGAN	**62.84**	**1.36**

Table 6.6: Comparison of initial modal single-modal representations and multimodal embeddings obtained with different methods. For each method, precision at 10 and its respective standard deviation are reported.

alize crossmodal mappings but is also crucial for obtaining high-quality embeddings in the discriminator by competing with it and providing additional synthetic samples.

Compared to multimodal AEs, CGANs are computationally expensive to train (20h on a GPU, compared to a few hours on a CPU for BiDNN), even for small images, and require both modalities to be present. However, the discriminator of a CGAN provides multimodal embeddings that are greatly improved over the initial representation spaces or each modality. The results are shown in Table 6.6. Representations learned with a CGAN not only outperformed multimodal AEs but they also significantly ($p = 99.9\,\%$ with a single-sided t-test) outperformed the state-of-the-art BiDNN model and obtained 62.84%.

6.7.2 Crossmodal Visualizations

Interestingly, the generator network can also be used to visualize crossmodal mappings in video hyperlinking. The generator can straightforwardly create synthetic images given an embedding of the speech transcripts as input. Examples of images generated for real transcripts on a few video segments, as well as their real visual counterparts are displayed in Table 6.7. The generator can also be reversed by simply applying the transposed learned kernels in an inversed architecture and slicing the obtained vector to the correct length (removing the part that is typically mapped to the noise input). In this case, given an image, embeddings in the textual domain can be obtained. A few examples of that translation are shown in Table 6.8.

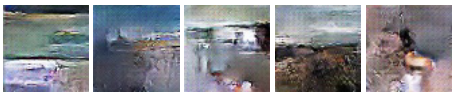
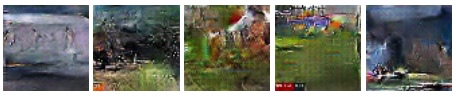
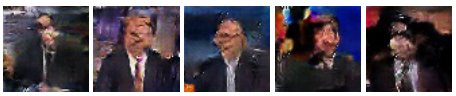
Input - Automatic Speech Transcript	Generated Images	Real Img.
"...insects emerged to take advantage of the abundance . the warm weather sees the arrival of migrant birds stone chests have spent the winter in the south ..."		
"...second navigation of the united kingdom . the north sea , it was at the north yorkshire moors between the 2 , starting point for the next leg of our journey along the coast ..."		
"...this is a dangerous time for injuries for athletes . having said that , some of these upbeat again a game . there she is running strongly she looks more comfortable ..."		
"...the role of my squadron afghanistan is to provide the the reconnaissance capability to use its or so forgave so using light armor of maneuvering around the area of ..."		

Table 6.7: Visualization of generated synthetic images given automatic transcripts (in lowercase, with some stop words removed) as input. In the last column, a real image from the video segment corresponding to the input automatic transcript is shown. CGANs provide good visualizations of the video hyperlinking model: in the last row, given speech transcripts related to war thematic, the model is expecting a news presenter, while the actual video segment contains footage from the war zone.

Input Image	Top Words in the Speech Modality
	britain, protecting, shipyard, carriers, jobs, vessels, current, royal, aircraft, securing, critics, flagships, foreclosures, economic, national
	north, central, rain, northern, eastern, across, scotland, southwest, west, north-east, northeast, south, affecting, england, midlands
	pepper, garlic, sauce, cumin, chopped, ginger, tomatoes, peppers, onion, crispy, parsley, grated, coconuts, salt, crust
	mountains, central, foreclosures, ensuing, across, scotland, norwegian, england, country, armor, doubting, migration, britain, southern

Table 6.8: Visualization of the top words in the representation space of automatic transcripts, given an input image.

6.8 Conclusion

In this chapter, we analyzed different ways of performing multimodal retrieval and more specifically video hyperlinking. We started by evaluating ways to obtain initial single-modal representation for speech transcripts in Section 6.4. After that, we progressed to multimodal fusion in Sections 6.5 and 6.6 where we first analyzed classical multimodal autoencoders and than bidirectional neural networks, a method we proposed to tackle a few downsides of classical multimodal autoencoders and obtain improved multimodal fusion by focusing on symmetric crossmodal translations. Given that crossmodal translations are typically implicit and not readily human-interpretable, in Section 6.7, we explored the possibility to perform multimodal fusion with a discriminative network of a conditional generative adversarial network while using the generative network to visualize learned crossmodal translations and synthesize images for a given speech transcript. Given the multitude of methods described in this section exploring the video hyperlinking task, we provide a short and clear summary of the analyzed methods that could serve as a generic guideline for future endeavors:

- We discourage the use of single-modal approaches and strongly encourage performing multimodal fusion.
- To improve multimodal fusion, it is better to focus on crossmodal translations first. Bidirectional deep neural networks (BiDNNs) consistently outperform classical multimodal autoencoders by a substantial margin and define the new state of the art both in terms of multimodal retrieval and video hyperlinking.
- Conditional generative adversarial networks (CGANs) seem to perform even better than BiDNNs as multimodal fusion methods and are additionally able to synthesize directly into the initial domain, which helps a human observer visualize the crossmodal translations learned by the model. However, CGANs are in their current form very hard to train and limited to very small image sizes. Training CGAN model with small images of 64×64 pixels takes about 20h on a NVIDIA Titan X GPU, compared to a few hours on a CPU for a BiDNN model trained with CNN embeddings that are taken from a significantly bigger image. While CGANs obtained better results on a dataset where images where scaled down (62.86% for CGANs compared to 59.66% for BiDNNs in terms of precision at 10), it is very easy to use bigger images with BiDNNs and obtain the state-of-the-art performance that they provide (80.00% in terms of precision at 10) in a fraction of the cost. Until new CGAN models, that are faster to train and can use bigger images, are evaluated and developed, the primary advantage of CGANs is their ability to visualize crossmodal translations.

Chapter 7

Assessing Diversity in Video Hyperlinking

Contents

In Chapter 6, we analyzed each method solely in regard of its ability to provide targets that are related to their anchors through a measure of similarity. Unfortunately, emphasizing relevance by rewarding highly similar content in terms of speech and visual features does not imply diversity in the set of targets that are proposed for a given anchor. This lack of diversity is considered as detrimental in many exploration scenarios, in particular when users' intentions and information needs are not known at the time of linking. In this case, providing relevant links that cover a number of possible extensions with respect to the anchor's content is desirable. Clearly, having a set of diverse targets strongly improves the chance for any user to find at least one interesting link to follow, whatever his/her initial intentions. Additionally, target diversity directly improves serendipity, i.e., unexpected yet relevant links, offering the possibility to drift from the initial anchor in terms of information so as to gain a better understanding of what can be found in the collection. Although it was proposed that diversity, together with relatedness, will be evaluated as part of TRECVID's video hyperlinking task [28] this is still currently not done and there is no notion on how popular systems perform in terms of diversity. To fill in this missing information and better understand diversity, how to evaluate it and how does it compare to relatedness in our methods, we decided to investigate the problem of diversity through live (a questionnaire) and intrinsic evaluations.

Topic 3	words	love, home, feel, life, baby
	visual concepts	singer, microphone, sax, concert, flute
Topic 7	words	food, bit, chef, cook, kitchen
	visual concepts	fig, acorn, pumpkin, guava, zucchini
Topic 25	words	years, technology, computer, key, future
	visual concepts	tape-player, computer, equipment, machine, appliance

Table 7.1: Three multimodal topics represented by their top-5 words and visual concepts

7.1 Video Hyperlinking with Bimodal LDA

In addition to bidirectional neural networks (BiDNN), one of the main contributions of this work and the current state of the art in video hyperlinking, we also evaluate bimodal latent Dirichlet allocation (BiLDA) - an older, not deep learning based, approach used in our team and proposed by Anca-Roxana Tudoran, a strong advocate of diversity in video hyperlinking [99, 10].

With a latent Dirichlet allocation (LDA), the similarity between two documents is measured via the similarity of the latent topics they share rather than by direct content comparison [9]. Recently, based on seminal work on multilingual topic modeling [102], multimodal extensions of LDA were proposed for crossmodal video hyperlinking [21], combining the potential for diversity offered by topic models and by multimodality. As for BiDNN, words extracted from the automatic transcripts and the visual concepts from the keyframes are used in the bimodal LDA (BiLDA).

The LDA model is based on the idea that latent variables, i.e., topics, which explain how words in documents have been generated, exist. Fitting such a generative model to a document means finding the best set of such latent variables in order to explain the observed data. As a result, documents are seen as mixtures of latent topics, while topics are probability distributions over words. The multimodal extension in [21] considers that each latent topic is defined by two probability distributions, one over each modality (or language in [102]). The BiLDA model is thus trained on parallel documents, assuming that the underlying topic distribution is common to the two modalities. In the case of videos, parallel documents are straightforwardly obtained by considering the transcripts and the visual concepts of a video segment as two parallel documents sharing the same underlying topic distribution.

Training, i.e., determining the topics from a given collection of videos, is achieved by Gibbs sampling, as for standard LDA [105], with the number of latent topics set to 700. Given a set of documents in the text (resp. visual) modality with vocabulary V_1 (resp. V_2), the probability that a word $w_i \in V_1$ (resp. visual concept $c_i \in V_2$) corresponds to topic z_j is estimated as

$$p(w_i|z_j) = \frac{n_{z_j}^{w_i} + \beta}{\sum_{x=1}^{|V_1|} n_{z_j}^{w_x} + \beta|V_1|} \tag{7.1}$$

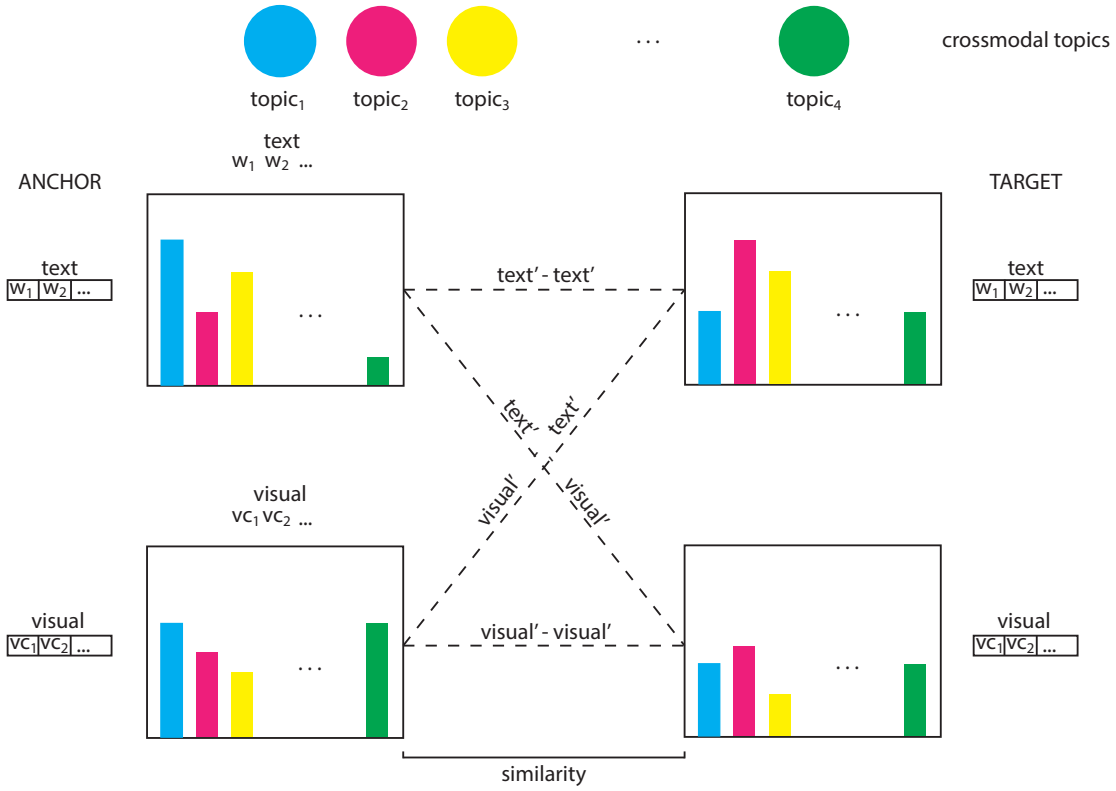


Figure 7.1: Illustration of the multimodal and crossmodal matching with the BiLDA model.

where $n_{z_j}^{w_i}$ is the number of times that topic z_j was assigned to word w_i in the training data and β is a Dirichlet prior.

This training step provides a mapping between topics of the two modalities by means of the topics. Table 7.1 displays examples of this mapping. For each topic, we show the 5 most probable words and visual concepts. Sometimes, words and visual concepts are directly related (e.g., *computer* in topic 25). However, the relation can be more subtle, as in topic 3 where visual concepts describe a stage, and words are utterances frequently encountered in the lyrics of songs.

The interest of topic models lies in the fact that video segments dealing with similar topics will tend to have similar distribution over the latent topics. This enables the indirect comparison of two video segments by comparing the distribution of latent topics, rather than using their multimodal content, thus potentially enabling a diversity of content (within documents from closely related topics). Formally, given a video segment d, the idea is to represent the segment as a vector collecting the topic probabilities:

$$p(d|z_j) = \left(\prod_{i=1}^{n_x} p(w_i|z_j) \right)^{1/n_x} \tag{7.2}$$

where n_x is the size of the vocabulary in d and w_i is the i^{th} word or visual concept in d. Note that $p(d|z_j)$ is an approximation of the posterior $p(z_j|d)$, considering a uniform distribution of topics, which is a reasonable assumption. The similarity score between any

two segments is given by a cosine similarity between their corresponding vectors after *L2*-normalization.

In practice, the probabilities $p(d|z_j)$ can be obtained from either one of the two modalities (using the corresponding distributions $p(\cdot|z_j)$), thus enabling multimodal and crossmodal matching as illustrated in Figure 7.1. In this chapter, we considered visual to text matching, representing the distribution of topics based on visual concepts for the anchor and on automatic transcripts for the targets.

7.2 Assessing Perceived Diversity

In addition to bimodal latent Dirichlet Analysis (BiLDA) and bidirectional deep neural networks (BiDNN) we evaluate a simple single-modal baseline. The three system we evaluate in terms of diversity are thus:

- A baseline consisting solely on automatic speech transcripts. The baseline system implements a bag-of-words representation for each segment with TF-IDF weighting [94] along with cosine similarity. Inverse document frequencies were estimated on the set of anchors plus the set of proposed targets.
- A bimodal latent Dirichlet allocation (BiLDA) method of performing video hyperlinking that was developed in the team used at TRECVID's 2015 video hyperlinking task [11].
- Bidirectional deep neural networks (BiDNN) that were previously explained and well analyzed in terms of anchor-target relatedness in this work.

To evaluate diversity, we created an online questionnaire, illustrated in Figure 7.3, where evaluators were presented with an anchor and 3 lists of 5 relevant targets, one for each method. They were asked to rank those lists from the least diverse (rank 1) to the most diverse (rank 3). In the evaluation interface, the anchor appeared on the top of the page, followed by 3 columns of 5 targets each, in a randomized order. Each segment was represented by a key image from which the video could be played, along with 10 keywords and 10 key concepts to facilitate the task, potentially avoiding the need to watch all 16 video segments thoroughly.

All the methods were trained on the TRECVID's 2015 video hyperlinking dataset that is basically the same as the dataset formed after TRECVID 2014 and described in Section 6.2, except that the provided groundtruth is different and contains different anchors and proposed targets.

A session consisted in ranking the lists for the 16 anchors selected, however not all evaluators completed their session. Since the order of anchors was also randomized per session, we kept all votes to report results on as many judgments as possible. In total, 25 persons, mostly from academia, participated in the evaluation, the vast majority of them not familiar with the video hyperlinking task. A total of 176 votes were recorded, with an average of 11 votes per anchor. The annotation took approximately 16 minutes to complete (median time), which corresponds to about one minute per anchor. Results are summarized in Figure 7.2 where the average rank is plotted (dot) for each method, with an error bar depicting

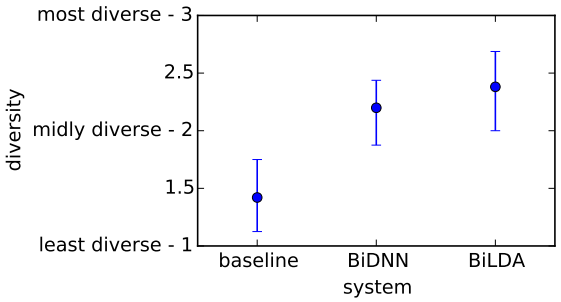


Figure 7.2: Average rank of systems with respect to diversity as perceived by evaluators

the dispersion of judgments among users—the lowest/highest average rank assigned to the method by a particular user.

Perceptive evaluations show a significant difference between the speech-transcript-only baseline (average rank of 1.42) and the BiLDA and BiDNN methods (average ranks of 2.20 for bidirectional deep neural networks and 2.38 bimodal latent Dirichlet analysis). It is also interesting to note that judgments are rather consistent across evaluators, for instance with average ranks from 1.12 to 1.75 for the baseline, confirming the ability of human evaluators to judge diversity.

BiLDA proposed targets were globally perceived as more diverse than those found by BiDNN (significant at $\alpha = 0.01$ according to a paired one-tailed t-test), even though BiLDA performs less than BiDNN in terms of relevance [10]. Bidirectional deep neural networks offer a very good compromise between relevance and diversity. Bimodal LDA offers better potential for diversity but weak performance in terms of relevance still appears as a limitation for this method.

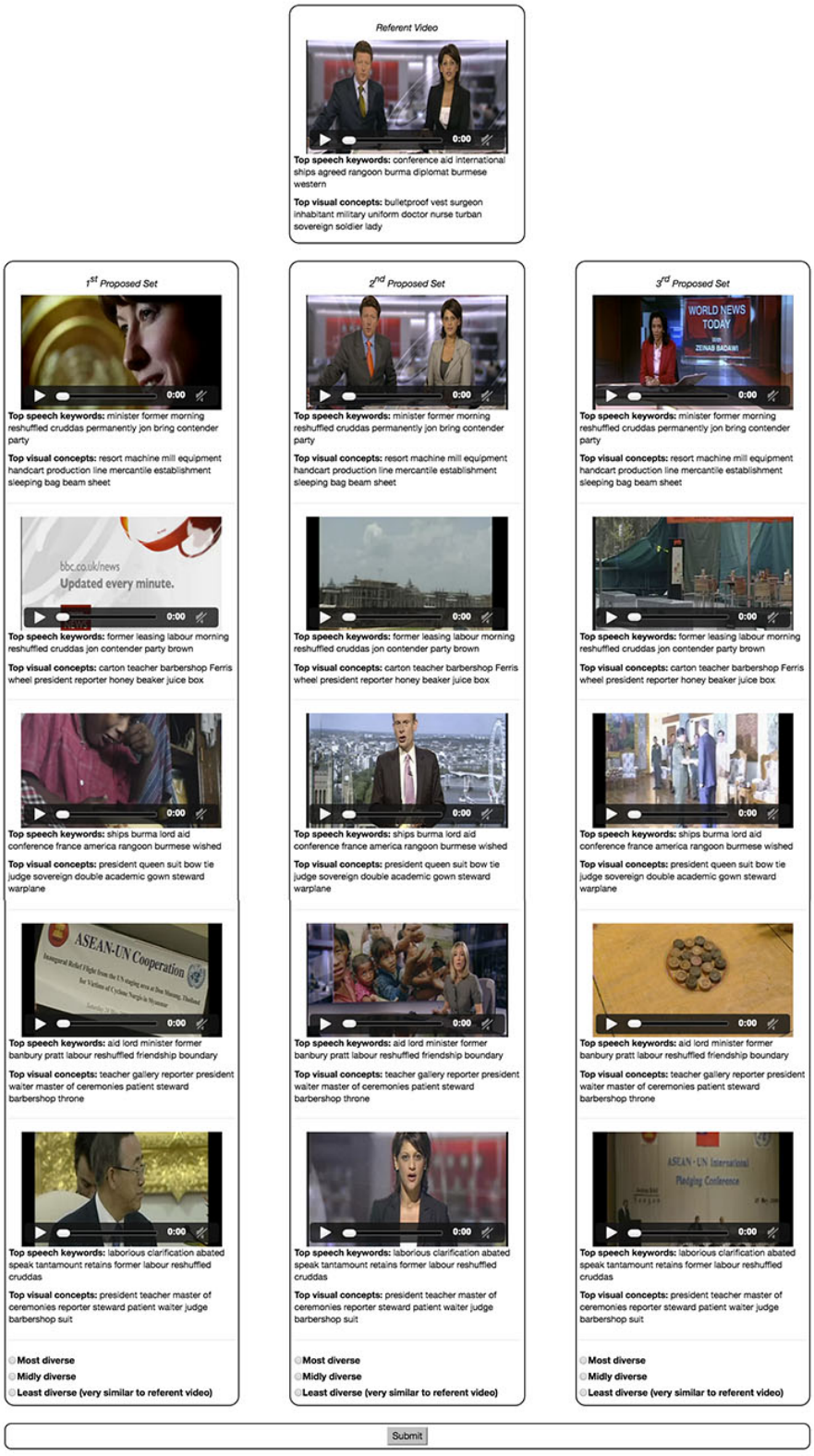


Figure 7.3: Evaluating diversity: the questionnaire presented two evaluators consisted of an anchor (reference video segment) presented on top and three columns (each for one system), containing 5 proposed targets (all relevant) each. Evaluators had to rank the three systems from "most diverse" to "least diverse".

7.3 Intrinsic Measures of Diversity

Perceptive evaluations, typically performed through online questionnaires, represent valid statistics based on a representative sample of the relevant population - the users of such systems. These evaluations are however expensive and time consuming as it is necessary not only to setup an evaluation platform but also to request the collaboration of a larger number of human evaluation. From the difficulty of performing such evaluations, the need for intrinsic and automatic evaluations arises. Together with Rémi Bois [10], we evaluate the feasibility of using intrinsic measures for evaluating diversity and we compare the obtained results to our previously described results, obtained through manual human evaluation.

Table 7.2 reports a number of intrinsic indicators of the diversity of a list of targets. n_u ($\in [10, 50]$) is the average number of unique key words/concepts in the top-5 relevant segments of an anchor, where the bigger n_u the better the diversity. A value of 10 indicates that all targets have the same key words/concepts. $\overline{d}_a$ is the average cosine similarity between the anchor and the top-5 relevant targets computed over the transcript or over the set of visual concepts. $\overline{d}_i$ measures the similarity within the top-5 targets of an anchor, computed as the average cosine similarity between any two pairs of targets in the top-5 list. In these last two cases, the larger the value, the less diverse the list of 5 targets.

Results in Table 7.2 clearly demonstrate that bidirectional deep neural networks offer a significantly greater diversity of relevant targets than the baseline. Diversity shows both from the lexical standpoint and from the visual one, where the difference between the baseline and crossmodal methods is stronger at the lexical level. Bidirectional deep neural networks appear to be slightly better than bimodal latent Dirichlet analysis in terms of average distance from targets to anchor as well as in terms of target dispersion.

	Transcripts			**Visual Concepts**		
	n_u	$\overline{d}_a$	$\overline{d}_i$	n_u	$\overline{d}_a$	$\overline{d}_i$
baseline	29.8	0.51	0.61	35.6	0.61	0.71
BiDNN	40.8	0.20	0.12	46.7	0.42	0.31
BiLDA	40.0	0.25	0.16	38.0	0.48	0.41

Table 7.2: Intrinsic indicators of the diversity of the top-5 relevant targets: number of unique keywords/concepts (n_u) average distance between targets and anchor ($\overline{d}_a$), average dispersion between targets ($\overline{d}_i$).

Perceptive evaluations by users confirm the results obtained with intrinsic evaluations, with a significant difference between the transcript-only baseline (average rank of 1.42) and the two crossmodal methods (average ranks of 2.20 and 2.38 for BiDNN and BiLDA respectively). These results indicate the feasibility of using evaluations based on intrinsic indicators either for preliminary comparisons of different methods or in an absence of a human evaluation framework.

7.4 Conclusion

Although relatedness of anchors and the targets proposed to them is currently the most extensively researched topic in video hyperlinking and precision is the only metric currently used at TRECVID's video hyperlinking evaluation, we emphasize the importance of the diversity within the proposed set of targets. To evaluate the diversity of the most used previously described methods, we run an online questionnaire that aimed at quantifying the human perception and, proved that a human evaluation can significantly estimate diversity and gave a scoring of the evaluated methods.

An important point we proved is the fact that diversity can be assessed not only using perceptual tests, but also using intrinsic dispersion measures. Intrinsic measures are easy to obtain and yield conclusions similar to the ones made with manual human evaluations, opening the door to large-scale studies on diversity in video hyperlinking.

We focus on crossmodal approaches for target selection in video hyperlinking as a mean to offer a diversity of targets. Intrinsic and perceptive evaluations show that crossmodal approaches are significantly better than a single-modal, speech transcripts based method in terms of diversity. Bidirectional deep neural networks (BiDNNs) offer a very good compromise between state-of-the-art relevance and diversity. Bimodal latent Dirichlet analysis (BiLDA) offers a better potential for diversity but weak performance in terms of relevance still appears as a limitation for this method. However, recent perceptual studies on LDA-derived targets show that combination of topic models can yield improved performances, also in terms of accuracy [99].

Chapter 8

Conclusion

Contents

In this last chapter, we first review the original objectives of this thesis and then progress to analyzing the contributions made within this work. In the last section, we will describe possible research leads that have been left unanswered within this manuscript but would be interesting to explore in followup works.

8.1 Thesis Objective

In this dissertation, we evaluate the thesis that neural embeddings are well suited for multimodal fusion. The objective of this work, entitled "deep neural architecture for automatic representation learning from multimedia multimodal data", was to evaluate existing and develop new methods for unsupervised representation learning of multimodal data in the context of multimedia. However, prior to doing that, we focused on simpler problems involving solely one modality.

The first objective was to evaluate architectures for single-modal textual and visual inputs. For text inputs, the goal was to compare word and text embedding methods and neural architectures for modeling sequences. For visual inputs, we set up the goal of predicting future motion given a single static image of a person performing a simple action. The goal was to evaluate existing methods and architectures, combining them, thus providing a potential improvement for each task and evaluating their feasibility for different subtasks in the more complex multimodal architectures developed next.

The second and primary objective was to develop and evaluate more complex deep learning architectures that make use of different input modalities, namely visual and textual, and perform multimodal fusion as well as crossmodal translation. The goal was to improve on multimodal retrieval methods by developing architectures that yield improved multimodal embeddings when fusing the two initially disjoint modalities. The main focus was put on video hyperlinking, a specific variation of multimodal retrieval where the main task is to retrieve a set of video segments that might be of interest to a viewer of a specific video segment. As video hyperlinking is a task evaluated as part of the TRECVID international benchmarking initiative, we also aimed at evaluating the methods we developed in a live setup through the TRECVID initiative. Last but not least, while developing methods to improve multimodal fusion, we also wanted to explore the possibility of visualizing the learned models in natural form from a human observer.

8.2 Summary of the Contributions

Regarding the first objective of evaluating single-modal architectures for modeling textual or visual inputs, we focused on two tasks. For textual modalities we evaluated different architectures in spoken language understanding or, more specifically, in the task of slot filling. We evaluated the performances of different recurrent neural architectures, from simple RNN architectures and their Jordan and Elman modifications to gated recurrent architectures that better model what information to retain or dismiss, such as LSTMs and GRUs. We have shown that the best way to model sequences is to use bidirectional GRU networks, that are simpler and also faster than LSTM networks. However, we have also shown that most of the gain from recurrent neural network based architectures in spoken language understanding comes from their initial text embedding (either learned jointly or separately with methods like *Word2Vec*) and not from their sequence modeling ability. While (gated) recurrent neural networks are well suited for modeling sequences, it seems that their inability to model output label dependencies prevents them from performing better than conditional random fields in the task of slot filling. Regarding visual modalities, we focused on action prediction and developed a simple architecture that predicts future motion after an arbitrary time difference, from a single image. We stretched the possibilities of convolutional neural networks in order to allow them to learn a representation of a person that is able to implicitly encode their stance and posture. Based on this representation, a "deconvolutional" network is able to synthesize a correct prediction of the person's motion, correctly anticipating the moving direction and change in posture, for an arbitrary, non-discretized, temporal displacement.

For the second objective, to improve multimodal fusion, we first developed a new architecture (bidirectional deep neural networks) that, contrary to existing multimodal autoencoders, focuses on crossmodal translations and creating a common representation space for the two (text to image and image to text) crossmodal translations. This new representation space is then used to perform multimodal fusion. We have shown on various offline evaluations and with different input representations for each modality that our method provides multimodal embeddings that significantly outperform embeddings obtained with multimodal autoencoders. In addition to offline evaluations, we also participated to the video hyperlinking evaluation, as part of the TRECVID international benchmarking initiative, where our method performed best and defines the state of the art today. We also ana-

lyzed our proposed method in terms of diversity through a web based, manual evaluation and determined that it offers good performances not only in terms of precision but also in terms of diversity. In the last part, we evaluated the possibility of using conditional generative adversarial networks for performing multimodal fusion, while also preserving the ability to synthesize into the original domain (the image domain) in order to offer visualizations of the learned crossmodal translations. We have shown that conditional generative adversarial networks can be used for multimodal fusion and that they do, for smaller image sizes, achieve state-of-the-art performance. However, they are currently very complex to train and limited in terms of input image sizes. For this reason, they cannot currently compete with either multimodal autoencoders or our proposed bidirectional deep neural networks but they can, however, offer nice visualizations of the learned crossmodal model.

8.3 Possible Future Work

Several possible future directions can be explored. Regarding spoken language understanding, more complex architectures that partially or fully model output level dependencies would have to be investigated. Such possible architectures can vary from simple added connections to previously decided labels to architectures that provide multiple output labels at once, with a modified cost function that models the output probabilities. In regards to multimodal fusion, especially when related to video hyperlinking, there are also multiple possible paths to explore. Bidirectional neural networks defined the new state of the art but are still rather simple. Exploring possible ways to either improve them or combine them with other methods could be rewarding. Possible ways of improving bidirectional neural networks could be either by adding simple modification like adding an additional cost function parameter that forces the two middle layers to be as similar as possible or by adding layers that would allow end-to-end learning. We expect an additional cost function to easily improve symmetry in the projections without throttling down the network with rigidity, the way introducing more shared variables would do. End-to-end learning is in general more optimal and can be fine-tuned at all levels and is thus expected to bring at least small improvements. More advanced ways of further improving bidirectional deep neural networks include but are not limited to exploring the possibility of using variational autoencoders [123], that have better statistical modeling properties, and plugging them as inputs to architectures designed to improve single modal retrieval. Generative adversarial networks are a very promising and fairly new research direction. While we have only shown their potential feasibility for video hyperlinking, they are currently being heavily improved. As architectures that are able to handle bigger image sizes are being developed, it would be interesting to evaluate whether they would at some point be able to achieve the same or better results as bidirectional neural networks in full scale. Additionally, we only evaluated a crossmodal translation that goes from speech representations to the image domain. It would be interesting to evaluate an end-to-end model that goes directly from speech to the image domain and back. This would be possible by adding a (gated) recurrent neural network on one side, instead of the normal speech representation input, while leaving the rest of the generator untouched. The opposite direction should also be easily modeled by using a convolutional neural network, together with random noise as input, and an RNN to generate realistically looking sentences for the given visual input. The performance of such architectures are currently left to explore.

Bibliography

[1] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: http://tensorflow.org/.

[2] Steven P Abney. "Parsing by chunks". In: *Principle-based parsing*. Springer, 1991, pp. 257–278.

[3] George Awad, Jonathan Fiscus, Martial Michel, David Joy, Wessel Kraaij, Alan F Smeaton, Georges Quénot, Maria Eskevich, Robin Aly, and Roeland Ordelman. "Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking". In: *Proceedings of TRECVID*. Vol. 2016. 2016.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[5] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. "Theano: new features and speed improvements". In: *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*. 2012.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features". In: *Computer vision–ECCV 2006* (2006), pp. 404–417.

[7] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *Neural Networks, IEEE Transactions on* 5.2 (1994), pp. 157–166.

[8] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. "Theano: a CPU and GPU math expression compiler". In: *Proceedings of the Python for scientific computing conference (SciPy)*. Vol. 4. Austin, TX. 2010, p. 3.

[9] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3 (2003), pp. 993–1022. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=944919.944937.

[10] Rémi Bois, Vedran Vukotić, Anca-Roxana Simon, Ronan Sicre, Christian Raymond, Pascale Sébillot, and Guillaume Gravier. "Exploiting Multimodality in Video Hyperlinking to Improve Target Diversity". In: *International Conference on Multimedia Modeling*. Springer, Cham. 2017, pp. 185–197.

[11] Rémi Bois, Anca-Roxana Şimon, Ronan Sicre, Guillaume Gravier, and Pascale Sébillot. "IRISA at TRECVid2015: Leveraging Multimodal LDA for Video Hyperlinking". In: *Proc. of TRECVID*. 2015.

[12] Hélène Bonneau-Maynard, Sophie Rosset, Christelle Ayache, Anne Kuhn, and Djamel Mostefa. "Semantic Annotation of the French Media Dialog Corpus". In: *InterSpeech*. Lisbon, 2005.

[13] Michal Campr and Karel Ježek. "Comparing Semantic Models for Evaluating Automatic Document Summarization". In: *Text, Speech, and Dialogue*. 2015.

[14] M Emre Celebi and Kemal Aydin. *Unsupervised Learning Algorithms*. Springer, 2016.

[15] Miriam Cha, Youngjune Gwon, and H. T. Kung. "Multimodal sparse representation learning and applications". In: *CoRR* abs/1511.06238 (2015).

[16] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2172–2180.

[17] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[18] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[19] François Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: *arXiv preprint arXiv:1610.02357* (2016).

[20] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[21] Anca-Roxana Şimon, Ronan Sicre, Rémi Bois, Guillaume Gravier, and Pascale Sébillot. "IRISA at TRECVid2015: Leveraging Multimodal LDA for Video Hyperlinking". In: *Proc. of TRECVID*. 2015.

[22] Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. "Expanding the scope of the ATIS task: the ATIS-3 corpus". In: *HLT*. Plainsboro, NJ, 1994, pp. 43–48. ISBN: 1-55860-357-3.

[23] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 886–893.

[24] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. "Front-end factor analysis for speaker verification". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4 (2011), pp. 788–798.

[25] Marco Dinarelli, Vedran Vukotic, and Christian Raymond. "Label-dependency coding in Simple Recurrent Networks for Spoken Language Understanding". In: *International Conference on Spoken Language Processing (Interspeech) 2017*. 2017.

[26] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. "Learning a deep convolutional network for image super-resolution". In: *European Conference on Computer Vision*. Springer. 2014, pp. 184–199.

[27] Jeffrey L Elman. "Finding structure in time". In: *Cognitive science* 14.2 (1990), pp. 179–211.

[28] Maria Eskevich, Martha Larson, Robin Aly, Serwah Sabetghadam, Gareth J. F. Jones, Roeland Ordelman, and Benoit Huet. "Multimodal Video-to-Video Linking: Turning to the Crowd for Insight and Evaluation". In: *Proc. of the 23rd International Conference on Multimedia Modeling*. 2017.

[29] Maria Eskevich, Robin Aly, David N. Racca, Roeland Ordelman, Shu Chen, and Gareth J.F. Jones. "The Search and Hyperlinking Task at MediaEval 2014". In: *Working Notes MediaEval Workshop*. 2014.

[30] Fangxiang Feng, Xiaojie Wang, and Ruifan Li. "Cross-modal retrieval with correspondence autoencoder". In: *ACM Intl. Conf. on Multimedia*. 2014, pp. 7–16.

[31] D F Fouhey and C L Zitnick. "Predicting object dynamics in scenes". In: *CVPR*. 2014, pp. 2019–2026.

[32] L A Gatys, A S Ecker, and M Bethge. "A neural algorithm of artistic style". In: *CoRR* (2015).

[33] Jean-Luc Gauvain, Lori Lamel, and Gilles Adda. "The LIMSI broadcast news transcription system". In: *Speech Communication* 37.1-2 (2002), pp. 89–108.

[34] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM". In: *Neural computation* 12.10 (2000), pp. 2451–2471.

[35] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[36] Camille Guinaudeau, Anca Roxana Simon, Guillaume Gravier, and Pascale Sébillot. "HITS and IRISA at MediaEval 2013: Search and hyperlinking task". In: *Working Notes MediaEval Workshop*. 2013.

[37] Stefan Hahn, Patrick Lehnen, Christian Raymond, and Hermann Ney. "A Comparison of Various Methods for Concept Tagging for Spoken Language Understanding". In: *Proceedings of the Language Resources and Evaluation Conference*. Marrakech, Morocco, 2008.

[38] Stefan Hahn, Marco Dinarelli, Christian Raymond, Fabrice Lefèvre, Patrick Lehnen, Renato De Mori, Alessandro Moschitti, Hermann Ney, and Giuseppe Riccardi. "Comparing Stochastic Approaches to Spoken Language Understanding in Multiple Languages". In: *IEEE Transactions on Audio, Speech and Language Processing* 19.6 (2011), pp. 1569–1583. DOI: 10.1109/TASL.2010.2093520.

[39] Yulan He and Steve Young. "Semantic Processing using the Hidden Vector State Model". In: *Computer Speech and Language* 19 (2005), pp. 85–106.

[40] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[41] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. "Online Tracking by Learning Discriminative Saliency Map with Convolutional Neural Network." In: *ICML*. 2015, pp. 597–606.

[42] D A Huang and K M Kitani. "Action-reaction: Forecasting the dynamics of human interaction". In: *ECCV*. Springer. 2014, pp. 489–504.

[43] Daniel Jiwoong Im, Chris Dongjoo Kim, Hui Jiang, and Roland Memisevic. "Generating images with recurrent adversarial networks". In: *arXiv preprint arXiv:1602.05110* (2016).

[44] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. "Aggregating local descriptors into a compact image representation". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 3304–3311.

[45] Lu Jiang, Shoou-I Yu, Deyu Meng, Yi Yang, Teruko Mitamura, and Alexander G Hauptmann. "Fast and accurate content-based semantic search in 100m internet videos". In: *Proceedings of the 23rd ACM international conference on Multimedia*. ACM. 2015, pp. 49–58.

[46] J Johnson, A Alahi, and L Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution". In: *CoRR* (2016).

[47] Michael I Jordan. "Serial order: A parallel distributed processing approach". In: *Advances in psychology* 121 (1997), pp. 471–495.

[48] D Kingma and J Ba. "Adam: A method for stochastic optimization". In: *CoRR* (2014).

[49] K M Kitani, B D Ziebart, J A Bagnell, and M Hebert. "Activity forecasting". In: *ECCV*. Springer. 2012, pp. 201–214.

[50] H S Koppula and A Saxena. "Anticipating human activities using object affordances for reactive robotic response". In: *PAMI* 38.1 (2016), pp. 14–29.

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[52] Taku Kudo and Yuji Matsumoto. "Chunking with Support Vector Machines". In: *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*. NAACL '01. Pittsburgh, Pennsylvania: Association for Computational Linguistics, 2001, pp. 1–8. DOI: 10.3115/1073336.1073361. URL: http://dx.doi.org/10.3115/1073336.1073361.

[53] Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. "Leveraging Sentence-level Information with Encoder LSTM for Natural Language Understanding". In: *arXiv preprint arXiv:1601.01530* (2016).

[54] John D. Lafferty, Andrew Mccallum, and Fernando C. N. Pereira. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *International Conference on Machine Learning*. San Francisco, CA, USA, 2001, pp. 282–289.

[55] Dana Lahat, Tülay Adali, and Christian Jutten. "Multimodal data fusion: an overview of methods, challenges, and prospects". In: *Proceedings of the IEEE* 103.9 (2015), pp. 1449–1477.

[56] T Lan, T C Chen, and S Savarese. "A hierarchical representation for future action prediction". In: *ECCV*. Springer. 2014, pp. 689–704.

[57] Antoine Laurent, Nathalie Camelin, and Christian Raymond. "Boosting bonsai trees for efficient features combination : application to speaker role identification". In: *InterSpeech*. Singapour, 2014.

[58] Thomas Lavergne, Olivier Cappé, and François Yvon. "Practical Very Large Scale CRFs". In: *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. Uppsala, Sweden: Association for Computational Linguistics, 2010, pp. 504–513. URL: http://www.aclweb.org/anthology/P10-1052.

[59] Quoc V Le and Tomas Mikolov. "Distributed Representations of Sentences and Documents." In: *ICML*. Vol. 14. 2014, pp. 1188–1196.

[60] Rémi Lebret, Joël Legrand, and Ronan Collobert. *Is Deep Learning Really Necessary for Word Embeddings?* Tech. rep. Idiap, 2013.

[61] Yann LeCun, LD Jackel, Leon Bottou, A Brunot, Corinna Cortes, JS Denker, Harris Drucker, I Guyon, UA Muller, Eduard Sackinger, et al. "Comparison of learning algorithms for handwritten digit recognition". In: *International conference on artificial neural networks*. Vol. 60. Perth, Australia. 1995, pp. 53–60.

[62] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. "Photo-realistic single image super-resolution using a generative adversarial network". In: *arXiv preprint arXiv:1609.04802* (2016).

[63] C Liu, J Yuen, and A Torralba. "Sift flow: Dense correspondence across scenes and its applications". In: *PAMI* 33.5 (2011), pp. 978–994.

[64] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3431–3440.

[65] David G Lowe. "Object recognition from local scale-invariant features". In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[66] Hungtsung Lu, Yuanming Liou, Hungyi Lee, and Linshan Lee. "Semantic Retrieval of Personal Photos Using a Deep Autoencoder Fusing Visual Features with Speech Annotations Represented as Word/Paragraph Vectors". In: *Annual Conf. of the Intl. Speech Communication Association*. 2015.

[67] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. "Adversarial autoencoders". In: *arXiv preprint arXiv:1511.05644* (2015).

[68] Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding". In: *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*. 2013, pp. 3771–3775. URL: http://www.isca-speech.org/archive/interspeech_2013/i13_3771.html.

[69] Krystian Mikolajczyk and Cordelia Schmid. "A performance evaluation of local descriptors". In: *IEEE transactions on pattern analysis and machine intelligence* 27.10 (2005), pp. 1615–1630.

[70] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality". In: *Advances in Neural Information Processing Systems*. 2013.

[71] Mehdi Mirza and Simon Osindero. "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784* (2014).

[72] R Mottaghi, H Bagherinezhad, M Rastegari, and A Farhadi. "Newtonian image understanding: Unfolding the dynamics of objects in static images". In: *CoRR* (2015).

[73] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. "Multimodal deep learning". In: *Intl. Conf. on Machine Learning*. 2011.

[74] Anh Nguyen, Jason Yosinski, Yoshua Bengio, Alexey Dosovitskiy, and Jeff Clune. "Plug & play generative networks: Conditional iterative generation of images in latent space". In: *arXiv preprint arXiv:1612.00005* (2016).

[75] A van den Oord, N Kalchbrenner, and K Kavukcuoglu. "Pixel Recurrent Neural Networks". In: *CoRR* (2016).

[76] A van den Oord, N Kalchbrenner, O Vinyals, L Espeholt, A Graves, and K Kavukcuoglu. "Conditional image generation with pixelcnn decoders". In: *CoRR* (2016).

[77] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. "Context encoders: Feature learning by inpainting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2536–2544.

[78] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation." In: *EMNLP*. Vol. 14. 2014, pp. 1532–1543.

[79] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M Álvarez. "Invertible Conditional GANs for image editing". In: (2016).

[80] Florent Perronnin and Christopher Dance. "Fisher kernels on visual vocabularies for image categorization". In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE. 2007, pp. 1–8.

[81] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. "Improving the fisher kernel for large-scale image classification". In: *Computer Vision–ECCV 2010* (2010), pp. 143–156.

[82] S L Pintea and J C van Gemert. "Making a Case for Learning Motion Representations with Phase". In: (2016).

[83] S L Pintea, J C van Gemert, and A W M Smeulders. "Déja vu". In: *ECCV*. Springer. 2014, pp. 172–187.

[84] A Radford, L Metz, and S Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *CoRR* (2015).

[85] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *ICLR 2016*. 2015.

[86] M Ranzato, A Szlam, J Bruna, M Mathieu, R Collobert, and S Chopra. "Video (language) modeling: a baseline for generative models of natural videos". In: *CoRR* (2014).

[87] Christian Raymond. *Bonzaiboost*. 2013. URL: http://bonzaiboost.gforge.inria.fr.

[88] Christian Raymond and Giuseppe Riccardi. "Generative and Discriminative Algorithms for Spoken Language Understanding". In: *InterSpeech*. Antwerp, Belgium, 2007, pp. 1605–1608.

[89] S Reed, Z Akata, X Yan, L Logeswaran, B Schiele, and H Lee. "Generative adversarial text to image synthesis". In: *CoRR* (2016).

[90] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. "Generative adversarial text to image synthesis". In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 3. 2016.

[91] M Ruder, A Dosovitskiy, and T Brox. "Artistic style transfer for videos". In: *CoRR* (2016).

[92] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[93] M. Saito and E. Matsumoto. "Temporal Generative Adversarial Nets". In: *CoRR* (2016).

[94] Gerard Salton and Michael J McGill. *Introduction to modern information retrieval*. McGraw-Hill, Inc., 1986.

[95] Robert E. Schapire and Yoram Singer. "BoosTexter: A boosting-based system for text Categorization". In: *Machine Learning* 39 (2000), pp. 135–168.

[96] C Schuldt, I Laptev, and B Caputo. "Recognizing human actions: a local SVM approach". In: *ICPR*. Vol. 3. IEEE. 2004, pp. 32–36.

[97] Mike Schuster and Kuldip K Paliwal. "Bidirectional recurrent neural networks". In: *Signal Processing, IEEE Transactions on* 45.11 (1997), pp. 2673–2681.

[98] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. "CNN features off-the-shelf: an astounding baseline for recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2014, pp. 806–813.

[99] Anca-Roxana Simon. "Semantic structuring of video collections from speech: segmentation and hyperlinking". PhD thesis. Université de Rennes 1, 2015.

[100] Anca-Roxana Simon, Ronan Sicre, Rémi Bois, Guillaume Gravier, and Pascale Sébillot. "IRISA at TrecVid2015: Leveraging Multimodal LDA for Video Hyperlinking". In: *TRECVid 2015 Workshop*. 2015.

[101] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[102] Wim De Smet and Marie-Francine Moens. "Cross-language linking of news stories on the web using interlingual topic modelling". In: *Proc. of ACM Workshop on Social Web Search and Mining*. 2009. DOI: 10.1145/1651437.1651447. URL: http://doi.acm.org/10.1145/1651437.1651447.

[103] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. "Semi-supervised recursive autoencoders for predicting sentiment distributions". In: *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics. 2011, pp. 151–161.

[104] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. "Striving for simplicity: The all convolutional net". In: *arXiv preprint arXiv:1412.6806* (2014).

[105] Mark Steyvers and Tom Griffiths. "Probabilistic topic models". In: *Handbook of Latent Semantic Analysis* 427.7 (2007), pp. 424–440.

[106] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[107] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[108] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2818–2826.

[109] M Tatarchenko, A Dosovitskiy, and T Brox. "Multi-view 3D Models from Single Images with a Convolutional Network". In: *ECCV*. Springer. 2016, pp. 322–337.

[110] Tatiana Tommasi, Tinne Tuytelaars, and Barbara Caputo. "A Testbed for Cross-Dataset Analysis". In: *CoRR* abs/1402.5923 (2014).

[111] Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. "What is left to be understood in ATIS?" In: *Spoken Language Technology Workshop (SLT), 2010 IEEE*. IEEE. 2010, pp. 19–24.

[112] Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. "Grammar as a foreign language". In: *Advances in Neural Information Processing Systems*. 2015, pp. 2755–2763.

[113] C Vondrick, H Pirsiavash, and A Torralba. "Anticipating the future by watching unlabeled video". In: *CoRR* (2015).

[114] C Vondrick, H Pirsiavash, and A Torralba. "Generating videos with scene dynamics". In: *NIPS*. 2016, pp. 613–621.

[115] Vedran Vukotic, Christian Raymond, and Guillaume Gravier. "A step beyond local observations with a dialog aware bidirectional GRU network for Spoken Language Understanding". In: *International Conference on Spoken Language Processing (Interspeech) 2016*. 2016.

[116] Vedran Vukotić, Christian Raymond, and Guillaume Gravier. "Bidirectional Joint Representation Learning with Symmetrical Deep Neural Networks for Multimodal and Crossmodal Applications". In: *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*. ACM. 2016, pp. 343–346.

[117] Vedran Vukotic, Christian Raymond, and Guillaume Gravier. "Generative Adversarial Networks for Multimodal Representation Learning in Video Hyperlinking". In: *ACM International Conference in Multimedia Retrieval (ICMR) 2017*. 2017.

[118] Vedran Vukotic, Christian Raymond, and Guillaume Gravier. "Is it time to switch to Word Embedding and Recurrent Neural Networks for Spoken Language Understanding?" In: *InterSpeech*. Dresde, Germany, 2015.

[119] Vedran Vukotić, Christian Raymond, and Guillaume Gravier. "Multimodal and crossmodal representation learning from textual and visual features with bidirectional deep neural networks for video hyperlinking". In: *Proceedings of the 2016 ACM workshop on Vision and Language Integration Meets Multimedia Fusion*. ACM. 2016, pp. 37–44.

[120] Vedran Vukotić, Silvia-Laura Pintea, Christian Raymond, Guillaume Gravier, and Jan Van Gemert. "One-Step Time-Dependent Future Video Frame Prediction with a Convolutional Encoder-Decoder Neural Network". In: *19th International Conference on Image Analysis and Processing (ICIAP)*. 2017.

[121] J Walker, A Gupta, and M Hebert. "Dense optical flow prediction from a static image". In: *ICCV*. 2015, pp. 2443–2451.

[122] J Walker, A Gupta, and M Hebert. "Patch to the future: Unsupervised visual prediction". In: *CVPR*. IEEE. 2014, pp. 3302–3309.

[123] J Walker, C Doersch, A Gupta, and M Hebert. "An uncertain future: Forecasting from static images using variational autoencoders". In: *ECCV*. Springer. 2016, pp. 835–851.

[124] Xiaolong Wang and Abhinav Gupta. "Generative image modeling using style and structure adversarial networks". In: *European Conference on Computer Vision*. Springer. 2016, pp. 318–335.

[125] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. "Towards ai-complete question answering: A set of prerequisite toy tasks". In: *arXiv preprint arXiv:1502.05698* (2015).

[126] "Word Embeddings through Hellinger PCA, author = R. Lebret and R. Collobert". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Association for Computational Linguistics, 2014, pp. 482–490.

[127] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. "Show, attend and tell: Neural image caption generation with visual attention". In: *International Conference on Machine Learning*. 2015, pp. 2048–2057.

[128] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. "Recurrent Neural Networks for Language Understanding". In: *InterSpeech*. Interspeech, 2013. URL: http://research.microsoft.com/apps/pubs/default.aspx?id=200236.

[129] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. "Spoken language understanding using long short-term memory neural networks". In: *IEEE Spoken Language Technology Workshop* (2014).

[130] Guangnan Ye, Yitong Li, Hongliang Xu, Dong Liu, and Shih-Fu Chang. "Eventnet: A large scale structured concept library for complex event detection in video". In: *Proceedings of the 23rd ACM international conference on Multimedia*. ACM. 2015, pp. 471–480.

[131] Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N Do. "Semantic Image Inpainting with Perceptual and Contextual Losses". In: *arXiv preprint arXiv:1607.07539* (2016).

[132] J Yuen and A Torralba. "A data-driven approach for event prediction". In: *ECCV*. Springer. 2010, pp. 707–720.

[133] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. "StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks". In: *arXiv preprint arXiv:1612.03242* (2016).

[134] Fang Zhao, Jiashi Feng, Jian Zhao, Wenhan Yang, and Shuicheng Yan. "Robust LSTM-Autoencoders for Face De-Occlusion in the Wild". In: *arXiv preprint arXiv:1612.08534* (2016).

[135] William Cohen Zhilin Yang Ruslan Salakhutdinov. "Multi-Task Cross-Lingual Sequence Tagging from Scratch". In: *arXiv*. 2016.

Appendix A

List of Publications

Accepted peer-reviewed publications in decreasing chronological order:

1. Vedran Vukotić, Christian Raymond, and Guillaume Gravier. "A Crossmodal Approach to Multimodal Fusion in Video Hyperlinking". In *IEEE MultiMedia Special Issue: Vision and Language Integration Meets Multimedia Fusion*. 2018.

2. Vedran Vukotić, Silvia-Laura Pintea, Christian Raymond, Guillaume Gravier and Jan C. van Gemert. "One-Step Time-Dependent Future Video Frame Prediction with a Convolutional Encoder-Decoder Neural Network". In *Intl. Conf. on Image Analysis and Processing*. 2017.

3. Marco Dinarelli, Vedran Vukotić and Christian Raymond. "Label-dependency coding in Simple Recurrent Networks for Spoken Spoken Language Understanding". In *Annual Conf. of the Intl. Speech Communication Association – Interspeech*. 2017.

4. Vedran Vukotić, Christian Raymond, and Guillaume Gravier. "Generative Adversarial Networks for Multimodal Representation Learning in Video Hyperlinking". In *ACM International Conference on Multimedia Retrieval*. 2017.

5. Rémi Bois, Vedran Vukotić, Anca-Roxana Simon, Ronan Sicre, Christian Raymond, and Guillaume Gravier. "Exploiting Multimodality in Video Hyperlinking to Improve Target Diversity". In *International Conference on Multimedia Modeling*. 2017.

6. Vedran Vukotić, Silvia-Laura Pintea, Christian Raymond, Guillaume Gravier, and Jan Van Gemert. "OneStep Time-Dependent Future Video Frame Prediction with a Convolutional Encoder-Decoder Neural Network". In *Netherlands Conference on Computer Vision*. 2016.

7. Vedran Vukotić, Christian Raymond, and Guillaume Gravier. "A step beyond local observations with a dialog aware bidirectional GRU network for Spoken Language Understanding". In A*nnual Conf. of the Intl. Speech Communication Association – Interspeech*. 2016.

8. Vedran Vukotić, Christian Raymond, and Guillaume Gravier. "Multimodal and Crossmodal Representation Learning from Textual and Visual Features with Bidirectional Deep Neural Networks for Video Hyperlinking". In *ACM Multimedia 2016 Workshop: Vision and Language Integration Meets Multimedia Fusion*. 2016.

9. Vedran Vukotić, Christian Raymond, and Guillaume Gravier. "Bidirectional Joint Representation Learning with Symmetrical Deep Neural Networks for Multimodal and Crossmodal Applications". In *ACM International Conference on Multimedia Retrieval*. 2016.

10. Vedran Vukotić, Christian Raymond, and Guillaume Gravier. "Is it time to switch to Word Embedding and Recurrent Neural Networks for Spoken Language Understanding?'.' In *Annual Conf. of the Intl. Speech Communication Association – Interspeech*. 2015.

Appendix B

Slot Filling Datasets

B.1 ATIS

The air travel information system (ATIS) task [22] is dedicated to provide flight information. Three values are used for each word, the word itself, a class to which the word might belong and the target label. There are 37 word classes in ATIS and they represent clusters of words that have the same meaning, e.g., *country_name*, *airport_name* etc. Every word utilized within the dataset that belongs to a cluster is replaced by the name of the cluster. The target label is then predicted by using the set of words and/or word classes, where available. The word classes are also used to model the appearance of relevant classes when modeling the dialog history. The training set consists of 4,978 sentences while the testing set consists of 893 sentences. There are 1,117 unique words and 85 possible target labels.

B.2 MEDIA

The research project MEDIA [12] evaluates different SLU models of spoken dialogue systems dedicated to provide tourist information. A 1,250 French dialogue corpus has been recorded by ELDA following a Wizard of Oz protocol: 250 speakers have each followed 5 hotel reservation scenarios. This corpus has been manually transcribed, then conceptually annotated according to a semantic representation defined within the project. We used three values for each word: the word itself, a class to which the word might belong and the target label. The classes of words are clusters to which multiple words belong. E.g. all city names used within the corpus belong to a *city_name* class. Most words do not belong to any specific class and are used as such. The target labels are again predicted from the words and/or word classes and the word classes are used to model the appearance of relevant classes in the dialog history.

Table B.1 shows an example message from the MEDIA corpus with only concept-value information. The first column contains the segment identifier in the message, the second column shows the chunks W^c supporting the concept c of the third column. In the fourth column the value of the concept c in the chunk W^c is displayed.

n	W^c	c	**value**
1	yes	answer	yes
2	the	RefLink	singular
3	hotel	BDObject	hotel
4	which	null	
5	price	object	payment-amount
6	is below	comparative-payment	below
7	fifty five	payment-amount-int	55
8	euros	payment-currency	euro

Table B.1: Example of message with concept+value information. The original French transcription is: "*oui l'hôtel dont le prix est inférieur à cinquante cinq euros*"

The MEDIA corpus is split into 3 parts. The first part (720 dialogues, 12,908 sentences) is used for training the models, the second (79 dialogues, 1,259 sentences) is used for cross-validation, and the third part (200 dialogues, 3,005 sentences) is used for testing.

INSA de RENNES
Service des Formations

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse:
Deep neural architectures for automatic representation learning from multimedia multimodal data

Nom Prénom de l'auteur : VUKOTIC VEDRAN

Membres du jury :
- Madame LARSON Martha
- Monsieur HUET Benoît
- Monsieur QUENOT Georges
- Monsieur GRAVIER Guillaume
- Monsieur RAYMOND Christian
- Madame TUYTELAARS Tinne

Président du jury :

Date de la soutenance : 26 Septembre 2017

Reproduction de la these soutenue

☒ Thèse pouvant être reproduite en l'état
☐ Thèse pouvant être reproduite après corrections suggérées

Fait à Rennes, le 26 Septembre 2017

Signature du président de jury

Le Directeur,

M'hamed DRISSI

Résumé

La thèse porte sur le développement d'architectures neuronales profondes permettant d'analyser des contenus textuels ou visuels, ou la combinaison des deux. De manière générale, le travail tire partie de la capacité des réseaux de neurones à apprendre des représentations abstraites (descripteurs) de contenus multimédias de manière supervisée ou non, e.g., par des architectures de type auto-encodeur. Les principales contributions de la thèse sont les suivantes :

- Réseaux récurrents pour la compréhension de la parole (tâche dite de slot filling) : différentes architectures de réseaux sont comparées pour cette tâche (GRU, LSTM, CRF, etc.) sur leurs facultés à modéliser les observations ainsi que les dépendances sur les étiquettes à prédire; ces comparaisons systématiques sur 2 benchmarks montrent notamment que les conclusions actuelles de la littérature tirées à partir d'un seul benchmark sont en partie erronées.

- Prédiction d'image et de mouvement : nous proposons une architecture permettant d'apprendre une représentation d'une image représentant une action humaine afin de prédire l'évolution du mouvement dans une vidéo ; l'originalité du modèle proposé réside dans sa capacité à prédire des images à une distance arbitraire dans une vidéo, par opposition aux approches classiques qui ne permettent que la prédiction de l'image suivante.

- Encodeurs bidirectionnels multimodaux : le résultat majeur de la thèse concerne la proposition d'un réseau bidirectionnel permettant de traduire une modalité en une autre, offrant ainsi la possibilité de représenter conjointement plusieurs modalités (langage et image dans la thèse) ou de passer de l'une à l'autre ; l'approche a été étudiée principalement en recherche d'information multi/cross-modale et en structuration de collections de vidéos, dans le cadre d'évaluations internationales où l'approche proposée s'est imposée comme l'état de l'art.

- Réseaux adverses pour la fusion multimodale : poursuivant la quête de représentations multimodales la thèse propose d'utiliser les architectures génératives adverses pour apprendre des représentations multimodales ; des résultats préliminaires montrent que, sous certaines conditions notamment concernant la taille des images, ces approches s'avèrent plus efficaces que les encodeurs bidirectionnels tout en offrant la possibilité de visualiser les représentations dans l'espace des images.

Abstract

In this dissertation, the thesis that deep neural networks are suited for analysis of visual, textual and fused visual and textual content is discussed. This work evaluates the ability of deep neural networks to learn automatic multimodal representations in either unsupervised or supervised manners and brings the following main contributions:

- Recurrent neural networks for spoken language understanding (slot filling): different architectures are compared for this task (GRU, LSTM, CRF, etc.) with the aim of modeling both the input context and output label dependencies. Extensive systematic comparisons on two benchmarks are performe indicating that multiple results reported in literature and evaluated solely on dataset are often statistically insignificant and thus partially erroneous.

- Action prediction from single images: we propose an architecture that allow us to predict human actions from a single image. The architecture is evaluated on videos, by utilizing solely one frame as input. Contrary to classical approaches, our proposed architecture allows the generation of prediction at arbitrary temporal differences.

- Bidirectional multimodal encoders: the main contribution of this thesis consists of neural architecture that translates from one modality to the other and conversely and offers and improved multimodal representation space where the initially disjoint representations can translated and fused. This enables for improved multimodal fusion of multiple modalities (text and images in this work). The architecture was extensively studied and evaluated in international benchmarks within the task of video hyperlinking where it defined the state of the art today.

- Generative adversarial networks for multimodal fusion: continuing on the topic of multimodal fusion, we evaluate the possibility of using conditional generative adversarial networks to learn multimodal representations. Preliminary results indicate that such representations could be superior to representations obtained with multimodal autoencoders but that such neural architectures are currently limited in regards of image size. In addition to providing multimodal representations, generative adversarial networks permit to visualize the learned model directly in the image domain.

INSA RENNES

N° d'ordre : 17ISAR 24 / D17 - 24

Institut National des Sciences Appliquées de Rennes
20, Avenue des Buttes de Coëmes • CS 70839 • F-35708 Rennes Cedex 7
Tel : 02 23 23 82 00 - Fax : 02 23 23 83 96