Fusion Techniques for Named Entity Recognition and Word Sense Induction and Disambiguation

Edmundo-Pavel Soriano-Morales, Julien Ah-Pine, and Sabine Loudcher

Université de Lyon, Lyon 2, ERIC EA 3083, Bron, France,
{edmundo.soriano-morales, julien.ah-pine,
sabine.loudcher}@univ-lyon2.fr

Abstract. In this paper we explore the use of well-known multimodal fusion techniques to solve two prominent Natural Language Processing tasks. Specifically, we focus on solving Named Entity Recognition and Word Sense Induction and Disambiguation by applying feature-combination methods that have already shown their efficiency in the multimedia analysis domain. We present a series of experiments employing fusion techniques in order to combine textual linguistic features. Our intuition is that by combining different types of features we may find semantic relatedness among words at different levels and thus, the combination (and recombination) of these levels may yield gains in terms of metrics' performance. To our knowledge, employing these techniques has not been studied for the tasks we address in this paper. We test the proposed fusion techniques on three datasets for named entity recognition and one for word sense disambiguation and induction. Our results show that the combination of textual features indeed improves the performance compared to single feature representation and the trivial feature concatenation.

1 Introduction

Named Entity Recognition (NER) and Word Sense Induction and Disambiguation (WSI/WSD) requires textual features to represent the similarities between words in order to discern between different words' meanings. NER goal is to automatically discover, within a text, mentions that belong to a well-defined semantic category. The classic task of NER involves detecting entities of type Location, Organization, Person and Miscellaneous. The task is of great importance for more complex NLP systems, e.g, relation extraction, opinion mining. Common solutions to NER consist on one of the following: via matching patterns created manually or extracted semi-automatically; or by training a supervised machine learning algorithm with large quantities of annotated text. The latter being the currently more popular solution to this task.

Word Sense Induction and Disambiguation involves two closely related tasks[1]. WSI aims to automatically discover the set of possible senses for a target word given a text corpus containing several occurrences of said target word. Meanwhile, WSD takes a set of possible senses and determines the most appropriate sense for each instance of

[1] Even though these tasks are related, they are independent from one another. Still, in this paper we consider them to be a single one.

the target word according to the instance's context. WSI is usually approached as an unsupervised learning task, i.e., a cluster method is applied to the words occurring in the instances of a target word. The groups found are interpreted as the senses of the target word. The WSD task is usually solved with knowledge-based approaches, based on WordNet; or more recently with supervised models which require annotated data.

As stated before, both tasks rely on features extracted from text. Usually, these representations are obtained from the surrounding context of the words in the input corpus. Mainly two types of representations are used. According to their nature we call these features lexical and syntactical. The first type requires no extra information than that contained already in the analyzed text itself. It consists merely on the tokens surrounding a word, i.e., those tokens that come before and after within a fixed window. The second type, syntactical features, is similar to the lexical representation in that we also consider as features the tokens that appear next to the corpus' words. Nonetheless, it requires a deeper degree of language understanding. In particular, these features are based on part of speech tags, phrase constituents information, and syntactical functionality between words, portrayed by syntactical dependencies. Likewise, specific features, particular to a task are also employed. These features later on become standard features in the literature.

Most of the approaches in the literature dealing with these tasks use each of these features independently or stacked together, i.e., different feature columns in an input representation space matrix. In the latter case, features are usually combined without regards to their nature.

The main intuition of the present work is that word similarities may be found at different levels according to the type of features employed. In order to exploit these similarities, we look into multimedia fusion methods. In order to better perform an analysis task, these techniques combine multimodal representations, their corresponding similarities, or the decisions coming from models fitted with these features. In this paper, we try to mutually complement independent representations by utilizing said fusion techniques to combine (or fuse) features in the hope of improving the performance of the tasks at hand, specially compared to the use of features independently.

Fusion techniques have previously shown their efficiency, mainly on text and image related tasks, where there is a need to model the relation between images and text extracts. Here, in order to apply multimedia fusion techniques, we consider textual features as different modalities, i.e., instead of having textual and image features we have lexical and syntactical features. The main contribution of this work is to assess the effectiveness of simple yet untested techniques to combine classical and easy to obtain textual features. As a second contribution, we propose a series of feature combination and recombination to attain better results. We test our intuitions on both NER and WSI/WSD tasks and over four different corpora: CoNLL-2003 [17], WikiNER and Wikigold [4] for NER; Semeval-2007 [1] for WSI/WSD.

The rest of the paper is organized as follows: in Section 2, we go into further details about fusion techniques. We introduce the fusion operators that we use in our experiments in Section 3. Then, in Section 4 we show the effectiveness of the presented methods by testing them on NER and WSI/WSD and their respective datasets. Finally, in Section 5 we present our conclusions and future directions to explore.

2 Background and Related Work

In this section, we describe the fusion techniques we use in our methodology as well as relevant use-cases where they have been employed.

2.1 Multimodal Fusion Techniques

Multimodal fusion is a set of popular techniques used in multimedia analysis tasks. These methods integrate multiple media features, the affinities among these attributes or the decisions obtained from systems trained with said features, to obtain rich insights about the data being used and thus to solve a given analysis task [2, 3]. We note that these techniques come at the price of augmenting the training time of a system by increasing both the dimension space and/or the density of a given feature matrix.

In the multimodal fusion literature we can discern two main common types of techniques: early fusion and late fusion.

Early Fusion This technique is the most widely used fusion method. The principle is simple: we take both modal features and concatenate them into a single representation matrix. More formally, we consider two matrices that represent different modality features each over the same set of individuals. To perform early fusion we concatenate them column-wise, such that we form a new matrix having the same number of lines but increasing the number of columns to the sum of the number of columns of both matrices. The matrices may also be weighted as to control the influence of each modality.

The main advantage of early fusion is that a single unique model is fitted while leveraging the correlations among the concatenated features. The method is also easy to integrate into an analysis system. The main drawback is that we increase the representation space and may make it harder to fit models over it.

Late Fusion In contrast to early fusion, in late fusion the combination of multimodal features are generally performed at the decision level, i.e., using the output of independent models trained each with an unique set of features [5]. In this setting, decisions produced by each model are combined into a single final result set. The methods used to combine preliminary decisions usually involve one of two types: rule-based (where modalities are combined according to domain-specific knowledge) or linear fusion (e.g., weighting and then adding or multiplying both matrices together). This type of fusion is very close to the so-called ensemble methods in the machine learning literature. Late fusion combines both modalities in the same semantic space. In that sense, we may also combine modalities via an affinity representation instead of final decision sets. In other words, we can combine two modality matrices by means of their respective similarities. A final representation is then usually obtained by adding the weighted similarity matrices.

The advantages of late fusion include the combination of features at the same level of representation (either the fusion of decisions or similarity matrices). Also, given that independent models are trained separately, we can chose which algorithm is more adequate for each type of features.

Cross-media Similarity Fusion A third type of fusion technique, cross-media similarity fusion (or simply cross fusion), introduced in [2, 5], is defined and employed to propagate a single similarity matrix into a second similarity matrix. In their paper, the authors propagated information from textual media towards visual media. In our case, we transfer information among textual features. For example, to perform a cross fusion between lexical and syntactical features, we perform the following steps:

1. Compute the corresponding similarity matrices for each type of feature.
2. Select only the k-nearest neighbor for each word within the lexical similarity matrix. These neighbors are to be used as lexical representatives to enrich the syntactical similarities.
3. Linearly combine both similarity matrices (lexical k-nearest lexical neighbors with the syntactical features) via a matrix product.

Cross fusion aims to bridge the semantic gap between two modalities by using the most similar neighbors as proxies to transfer valuable information from one modality onto another one. Usually, the result of a cross fusion is combined with the previous techniques, early and late fusion. In this paper we perform experiment in that sense.

Hybrid Fusion We may leverage the advantages of the previous two types of fusion techniques by combining them once more in a hybrid setting. As described in [3, 18], the main idea is to simultaneously combine features at the feature level, i.e., early fusion, and at the same semantic space or decision level. Nonetheless, they define a specific type of hybrid fusion. In this paper, we adopt a looser definition of hybrid fusion. That is, we perform hybrid fusion by leveraging the combination of the fusion strategies described before.

We consider the first three types of fusion techniques (early fusion, late fusion and cross fusion) as the building blocks to the experiments we conduct. While we work with a single modality, i.e., textual data, we consider the different kinds of features extracted from it as distinct modalities. Our intuition being that the semantic similarities among words in these different spaces can be combined in order to exploit the latent complementarity between the lexical and syntactical representations. The fusion should therefore improve the performance of the NLP tasks at hand, NER and WSI/WSD.

Our first goal is to assess the effectiveness of the classic fusion methods and then, as a second goal, to propose new combinations that yield better outcomes in terms of performance than the simpler approaches. The new combinations are found empirically. Nonetheless, as we will show, their effectiveness replicates across different datasets and NLP tasks.

2.2 NER and WSI/WSD

To the best of our knowledge, there is no work that addresses both NER and WSI/WSD explicitly while using fusion techniques from the multimedia analysis domain. Still, we base our experiments on those carried on in [6, 8, 10] using well-known supervised (structured perceptron) and unsupervised (spectral clustering) learning algorithms. A thorough review on NER and WSI/WSD can be found in [13] and [14], respectively.

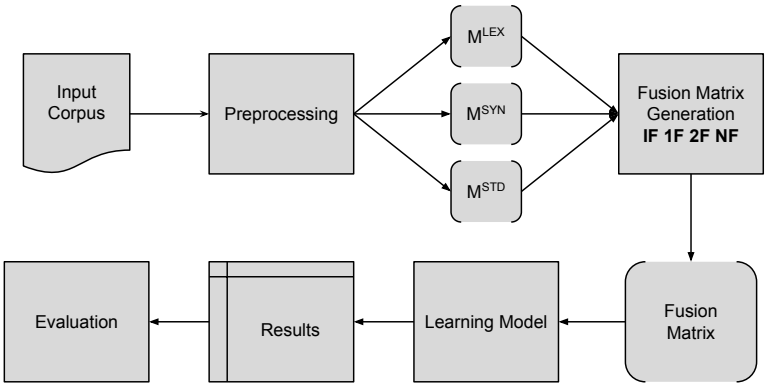

Fig. 1. Steps followed on our experiments. First the corpus is preprocessed, then features are extracted from the text. A fusion matrix is generated, which in turn is used as input to a learning algorithm. Finally, the system yields its results and to be analyzed.

3 Methodology

In the present section we address the core of the work performed in this paper. We formally describe the fusion techniques we employ in the next section. Also, we delineate the procedure followed in our experiments.

The experiments we carry on consist in generating fusion matrices that will serve as input to a learning algorithm in order to solve NER and WSI/WSD. These input feature matrices are based upon lexical, syntactical, or other types of representation. The procedure can be seen in Figure 1.

3.1 Fusion Strategies

We begin by presenting a formal definition of the fusion techniques employed and described in the previous sections. We define (weighted) early fusion, late fusion and cross fusion as follows:

Early Fusion

$$E(A, B) = \mathbf{hstack}(A, B) \tag{1}$$

Weighted Early Fusion

$$wE_{\alpha}(A, B) = \mathbf{hstack}(\alpha \cdot A, (1 - \alpha) \cdot B) \tag{2}$$

Late Fusion

$$L_{\beta}(A, B) = \beta \cdot A + (1 - \beta) \cdot B \tag{3}$$

Cross fusion

$$X_\gamma(A, B) = \mathbf{K}(A, \gamma) \times B \tag{4}$$

Parameters A and B are arbitrary input matrices. They may initially represent, for example, the lexical (M^{LEX}) or syntactical based (M^{SYN}) features matrix, or their corresponding similarity matrices, S^{LEX} and S^{SYN}, respectively. In a broader sense, matrices A and B may represent any pair of valid[2] fusion matrices.

In early fusion, $E(A, B)$, the matrices A and B are combined together via a function called $\mathbf{hstack}$ which concatenates, column-wise, both matrices A and B. Weighted early fusion represents the same operation as before with an extra parameter: α, which controls the relative importance of each matrix. In the following, we refer to both operations as early fusion. When α is determined, we refer to weighted early fusion.

Regarding late fusion $L_\beta(A, B)$, the β parameter determines again the importance of the matrix A, and consequently also the relevance of matrix B.

In cross fusion $X_\gamma(A, B)$, the $\mathbf{K}(\cdot)$ function keeps the top-γ closest words (columns) to each word (lines) while the rest of the values are set to zero.

Using the previously defined operators, we distinguish four levels of experiments:

1. **Single Features**: in this phase we consider the modalities independently as input to the learning methods. For instance, we may train a model for NER using only the lexical features matrix M^{LEX}.
2. **First Degree Fusion**: we consider the three elementary fusion techniques by themselves (early fusion, late fusion, cross fusion) without any recombination. These experiments, as well as those from the previous level, serve as the baselines we set to surpass in order to show the efficacy of the rest of the fusion approaches. As an example, we may obtain a representation matrix by performing an early fusion between the lexical matrix and the syntactical features matrix: $E(M^{LEX}, M^{SYN})$. In this level we distinguish two types of cross fusion: Cross Early Fusion (XEF) and Cross Late Fusion (XLF). The first one combines a similarity matrix with a feature matrix: $X(S^{LEX}, M^{SYN})$. The second one joins a similarity matrix with a similarity matrix: $X(S^{SYN}, S^{LEX})$.
3. **Second Degree Fusion**: we recombine the outputs of the previous two levels with the elementary techniques. This procedure then yields a recombination of "second-degree" among fusion methods. We introduce the four types of second degree fusions in the following list. Each one is illustrated with an example:
   (a) Cross Late Early Fusion (XLEF): $X(X(S^{STD}, S^{SYN}), M^{STD})$
   (b) Cross Early Early Fusion (XEEF: $X(S^{STD}, X(S^{STD}, S^{SYN}))$
   (c) Early Cross Early Fusion (EXEF): $E(M^{STD}, X(S^{LEX}, M^{STD}))$
   (d) Late Cross Early Fusion (LXEF): $L(M^{STD}, X(S^{STD}, M^{STD}))$
4. **N-Degree Fusion**: in this last level we follow a similar approach to the previous level by combining the output of the second-degree fusion level multiple times (more than two times) with other second-degree fusion outputs. Again, in this level we test the following two fusion operations:
   (a) Early Late Cross Early Fusion (ELXEF): $E(M^{STD}, L(M^{STD}, X(S^{STD}, M^{STD})))$
   (b) Early ELXEF (EELXEF): $E(M^{LEX}, E(E(M^{STD}, L(M^{STD}, X(S^{STD}, M^{STD}))), L(M^{LEX}, X(S^{SYN}, M^{LEX}))))$

[2] Valid in terms of having compatible shapes while computing a matrix sum or multiplication.

3.2 Feature Matrices

In the previous subsection we presented the fusion operators used in our experiments. Below we detail the three types of features used to describe the words of each of the tested corpus.

Lexical Matrix (LEX) For each token in the corpus, we use a lexical window of two words to the left and two words to the right, plus the token itself. Specifically, for a target word w, its lexical context is $(w_{-2}, w_{-2}, w, w_{+1}, w_{+2})$. This type of context features is typical for most systems studying the surroundings of a word, i.e., using a distributional approach [11].

Syntactical Matrix (SYN) Based on the syntactic features used in [11, 15], we derive contexts based on the syntactic relations a word participates in, as well as including the part of speech (PoS) of the arguments of these relations. Formally, for a word w with modifiers $m_1, \ldots, m_k$ and their corresponding PoS tags $p_1^m, \ldots, p_k^m$; a head h and its corresponding PoS tag p_h, we consider the context features $(m_1, p_{m_1}, lbl_1), \ldots, (m_k, p_{m_k}, lbl_k), (h, p_h, lbl\_inv_h)$. In this case, lbl and lbl_{inv} indicate the label of the dependency relation and its inverse, correspondingly. Using syntactic dependencies as features should yield more specific similarities, closer to synonymy, instead of the broader topical similarity found through lexical contexts.

NER Standard Features Matrix (STD) The features used for NER are based on those used in [8, 4]. The feature set consists of: the word itself, whether the word begins with capital letter, prefix and suffix up to three characters (within a window of two words to the left and two words to the right), and the PoS tag of the current word. These features are considered to be standard in the literature. We note that the matrix generated with these features is exclusively used in the experiments regarding NER.

3.3 Learning Methods

We use supervised and unsupervised learning methods for NER and WSI/WSD respectively. On the one hand, for NER, as supervised algorithm, we use an averaged structured perceptron [6, 8] to determine the tags of the named entities. We considered Logistic Regression and linear SVM. We chose the perceptron because of its performance and its lower training time.

On the other hand, for WSD/WSI, specifically for the induction part, we applied spectral clustering, as in [10], on the input matrices in order to automatically discover senses (a cluster is considered a sense). Regarding disambiguation, we trivially assign senses to the target word instances according to the number of common words in each cluster and the context words of the target word. In other words, for each test instance of a target word, we select the cluster (sense) with the maximum number of shared words with the current instance context.

4 Experiments and Evaluation

We experiment with four levels of fusion: Single Features (SF), First-degree Fusion (1F), Second-degree Fusion (2F) and N-degree Fusion (NF). The representation matrices for NER come from lexical context features M^{LEX}, syntactical context features M^{SYN} or standard features M^{STD}. On the other hand, experiments on WSI/WSD exclusively employ matrices M^{LEX} and M^{SYN}.

Our first goal is to compare the efficiency of the basic multimedia fusion techniques applied to single-modality multi-feature NLP tasks, namely NER and WSI/WSD. A second goal is to empirically determine a fusion combination setting able to leverage the complementarity of our features.

To this end, we evaluate the aforementioned 4 fusion levels. We note that the fusion combinations in the third and fourth level (2F and NF) are proposed based on the results obtained in the previous levels. In other words, in order to reduce the number of experiments, we restrict our tests to the best performing configurations. This is due to the large number of possible combinations (an argument to a fusion operation may be any valid output of a second fusion operation).

4.1 Named Entity Recognition

Pre-processing As is usual when preprocessing text before performing named entity recognition, [16], we normalize tokens that include numbers. This allows a degree of abstraction to tokens that contain years, phone numbers, etc.

Features The linguistic information we use are extracted with the Stanford's CoreNLP parser [12]. Again, the features used for these experiments on NER are those described before: lexical, syntactic and standard features, i.e., M^{LEX}, M^{SYN}, and M^{STD}, respectively.

Test Datasets We work with three corpora coming from two different domains:

(1) CoNLL-2003 (CONLL): This dataset was used in the language-independent named entity recognition CoNLL-2003 shared task [17]. It contains selected news-wire articles from the Reuters Corpus. Each article is annotated manually. It is divided in three parts: training (*train*) and two testing sets (*testa* and *testb*). The training part contains 219,554 lines, while the test sets contain 55,044 and 50,350 lines, respectively. The task was evaluated on the *testb* file, as in the original task.
(2) WikiNER (WNER): A more recent dataset [4] of selected English Wikipedia articles, all of them annotated automatically with the author's semi-supervised method. In total, it contains 3,656,439 words.
(3) Wikigold (WGLD): Also a corpus of Wikipedia articles, from the same authors of the previous corpus. Nonetheless, this was annotated manually. This dataset is the smaller, with 41,011 words. We used this corpus to validate human-tagged Wikipedia text. These three datasets are tagged with the same four types of entities: Location, Organization, Person and Miscellaneous.

Table 1. NER F-measure results using the Single Features over the three datasets. These values serve as a first set of baselines.

A	**Single Features**		
	CONLL	**WNER**	**WGLD**
M^{STD}	77.41	77.50	59.66
M^{LEX}	69.40	69.17	52.34
M^{SYN}	32.95	28.47	25.49

Evaluation Measures We evaluate our NER models following the standard CoNLL-2003 evaluation script. Given the amount of experiments we carried on, and the size constraints, we report exclusively the total F-measure for the four types of entities (Location, Organization, Person, Miscellaneous). WNER and WGLD datasets are evaluated on a 5-fold cross validation.

Results We present in this subsection the results obtained in the named entity recognition task, while employing the 4 levels of fusion proposed in the previous section.

In contrast to other related fusion works [2, 5, 9], we do not focus our analysis on the impact of the parameters of the fusion operators. Instead, we focus our analysis on the effect of the type of linguistic data being used and how, by transferring information from one feature type to another, they can be experimentally recombined to generate more complete representations.

Regarding the fusion operators' parameters, we empirically found the best configuration for β, from late fusion $L_\beta(A, B) = \beta \cdot A + (1 - \beta) \cdot B$, is $\beta = 0.5$. This implies that an equal combination is the best linear fusion for two different types of features.

In respect of the γ parameter, used in cross fusion $X_\gamma(A, B) = \mathbf{K}(A, \gamma) \times B$, we set $\gamma = 5$. This indicates that just few high quality similarities attain better results than utilizing a larger quantity of lower quality similarities.

Single Features Looking at Table 1, we see that the best independent features, in terms of F-measure come from the standard representation matrix M^{STD}. This is not surprising as these features, simple as they may be, have been used and proved extensively in the NER community. On the other hand, M^{LEX} performs relatively well, considering it only includes information contained in the dataset itself. Nevertheless, this representation that this kind of lexical context features are the foundation of most word embedding techniques used nowadays. While we expected better results from the syntactical features M^{SYN}, as they are able to provide not only general word similarity, but also functional, getting close to synonymy-level [11], we believe that the relatively small size of the datasets do not provide enough information to generalize

First Degree Fusion In Table 2 we present the First Degree fusion level. The best performance is obtained by trivially concatenating the representation matrices. This baseline proved to be the toughest result to beat. Late fusion does not perform well in this setting, still, we see further on that by linearly combining weighted representation

Table 2. NER F-measure results using first degree fusion (1F). B is either indicated on the table or specified as follows. Looking at EF, $\hat{b}_{EF} = E(M^{SYN}, M^{STD})$. In XEF, b^*_{XEF} takes the matrix from the set $\{M^{LEX}, M^{STD}\}$ which yields the best performing result. In XLF, $\hat{b}^*_{XLF}$ corresponds to the best performing matrix in $\{S^{LEX}, S^{SYN}\}$. These configurations serve as the main set of baseline results.

A	B			**Early Fusion**
		CONLL	**WNER**	**WGLD**
M^{LEX}	M^{SYN}	72.01	70.59	59.38
M^{LEX}	M^{STD}	78.13	79.78	61.96
M^{SYN}	M^{STD}	77.70	78.10	60.93
M^{LEX}	$\hat{b}_{EF}$	**78.90**	**80.04**	**63.20**
				Late Fusion
		CONLL	**WNER**	**WGLD**
S^{LEX}	S^{SYN}	**61.65**	58.79	44.29
S^{LEX}	S^{STD}	55.64	**67.70**	48.00
S^{SYN}	S^{STD}	50.21	58.41	**49.81**
				Cross Early Fusion
		CONLL	**WNER**	**WGLD**
S^{LEX}	M^{STD}	49.90	**70.27**	**62.69**
S^{SYN}	M^{STD}	47.27	51.38	48.53
S^{STD}	b^*_{XEF}	**52.89**	62.21	50.15
				Cross Late Fusion
		CONLL	**WNER**	**WGLD**
S^{LEX}	S^{STD}	27.75	**59.12**	38.35
S^{SYN}	b^*_{XLF}	36.87	40.92	39.62
S^{STD}	b^*_{XLF}	**41.89**	52.03	**39.92**

Table 3. NER F-measure results using second degree fusion (2F). In XLEF, a^* corresponds to the best performing matrix in the set $\{X(S^{STD}, S^{LEX}), X(S^{LEX}, S^{STD}), X(S^{STD}, S^{SYN})\}$. For XEEF, $\hat{b}_{XEEF} = E(M^{LEX}, M^{STD})$. In EXEF, b^*_{EXEF} takes the best performing matrix from $\{X(S^{SYN}, M^{LEX}), X(S^{LEX}, M^{LEX}), X(S^{LEX}, M^{STD}), X(S^{SYN}, M^{LEX}), X(S^{SYN}, M^{STD})\}$. Finally, in LXEF, $\hat{b}_{LXEF}$ takes the best possible matrix from $\{X(S^{LEX}, M^{STD}), X(S^{SYN}, M^{STD}), X(S^{SYN}, M^{LEX})\}$.

A	B			**Cross Late Early Fusion**
		CONLL	**WNER**	**WGLD**
$\hat{a}$	M^{STD}	37.69	**59.44**	**41.71**
$\hat{a}$	M^{LEX}	**38.31**	58.73	41.56
$\hat{a}$	M^{SYN}	29.31	52.06	34.91
				Cross Early Early Fusion
		CONLL	**WNER**	**WGLD**
S^{STD}	$\hat{b}_{XEEF}$	**54.34**	64.20	39.59
S^{LEX}	$\hat{b}_{XEEF}$	49.71	**71.84**	**45.14**
S^{SYN}	$\hat{b}_{XEEF}$	47.54	53.77	43.32
				Early Cross Early Fusion
		CONLL	**WNER**	**WGLD**
M^{STD}	b^*_{EXEF}	49.58	**77.32**	**61.69**
M^{LEX}	b^*_{EXEF}	49.79	66.22	53.54
M^{SYN}	b^*_{EXEF}	**51.53**	70.94	53.70
				Late Cross Early Fusion
		CONLL	**WNER**	**WGLD**
M^{STD}	$\hat{b}_{LXEF}$	54.82	**75.70**	**54.73**
M^{LEX}	$\hat{b}_{LXEF}$	**56.53**	62.27	52.39

matrices, we can add information to an already strong representation. Finally, regarding the cross fusion techniques, cross early and late fusion, we see that they depend directly on the information contained in the similarity matrices. We note that, as is the case on single features, the combinations with matrix S^{STD} yield almost always the best results. While these fusion techniques by themselves may not offer the best results, we see below that by recombining them with other types of fusion we can improve the general performance of a representation.

Second Degree Fusion The second degree fusion techniques presented in Table 3 show that the recombination of cross fusion techniques gets us closer to the early fusion baseline. With the exception of cross late early fusion, the rest of the recombination schemes yield interesting results. First, in cross early fusion, the best results, for the most part, are obtained while using the S^{LEX} matrix combined with the output of $E(M^{LEX}, M^{STD})$, which is still far from the baseline values. Concerning, EXEF, we get already close to surpass the baselines with the M^{STD} matrix, with the exception of the CONLL dataset. In LXEF, even though the cross fusion $X(S^{SYN}, M^{LEX})$ is not the best performing, we found experimentally that by combining it with M^{LEX} through a late fusion, it gets a strong complementary representation. Our intuition in this case was to complement M^{LEX} with itself but enriched with the S^{SYN} information. In the N-degree fusion results we discover that indeed this propagation of information helps us beat the baselines we set before.

N-degree Fusion Finally, the last set of experiments are shown in Table 4. Using a recombination of fusion techniques, a so-called hybrid approach, we finally beat the baselines (single features and early fusion) for each dataset. We note that the best configuration made use of a weighted early fusion with $\alpha = 0.95$. This indicates that the single feature matrix, M^{LEX} is enriched a small amount by the fusion recombination, which is enough to improve the results of said baselines. In CONLL, the early fusion (see Table 2) baseline being 78.13, we reached 78.69, the lowest improvement of the three datasets. Regarding the Wikipedia corpus, in WNER, we passed from 79.78 to 81.75; and in WGLD, from 61.96 to 67.29, the largest improvement of all. It is important that we tried the weighted Early Fusion operator with different α and the best result does not beat these fusion results.

In the next section we transfer the knowledge gained in this task to a new one, word sense induction and disambiguation.

4.2 Word Sense Induction and Disambiguation

Having learned the best fusion configuration from the previous task, in this experiments we set to test if the improvements achieved can be transfered into another NLP task, namely Word Sensed Induction and Disambiguation (WSI/WSD).

Pre-processing We simply remove stopwords and tokens with less than three letters.

Features We use the same set of features from the previous task, with the exception of the standard NER features, that is, those represented by M^{STD}, as they are specifically designed to tackle NER.

Test Dataset The WSI/WSD model is tested on the dataset of the Semeval-2007 WSID task [1]. The task was based on a set of 100 target words (65 nouns and 35 verbs), each word having a set of instances, which are specific contexts where the word appear. Senses are induced from these contexts and applied to each one of the instances.

Table 4. F-measure results using N-degree fusion (NF). In ELXEF, $\hat{b}_{ELXEF} = L(M^{LEX}, X(S^{SYN}, M^{LEX}))$. For EELXEF, $\hat{b}_{EELXEF} = E(E(M^{STD}, L(M^{LEX}, X(S^{SYN}, M^{LEX}))), L(M^{LEX}, X(S^{STD}, M^{LEX})))$ for CONLL and $\hat{b}_{EELXEF} = E(E(M^{STD}, L(M^{STD}, X(S^{SYN}, M^{STD}))), L(M^{LEX}, X(S^{SYN}, M^{LEX})))$ for WNER and WGLD. The best result is obtained in EELXEF when $\alpha = 0.95$.

A	B	**Early Late Cross Early Fusion**		
		CONLL	**WNER**	**WGLD**
M^{STD}	$\hat{b}_{ELXEF}$	67.16	79.45	62.37
		Early Early Late Cross Early Fusion		
		CONLL	**WNER**	**WGLD**
M^{LEX}	$\hat{b}_{EELXEF}$	65.01	78.02	62.34
$M^{LEX}_{\alpha=0.95}$	$\hat{b}_{EELXEF}$	**79.67**	**81.79**	**67.05**
EF Baseline		78.90	80.04	63.20

Evaluation Measures Being an unsupervised task, the evaluation metrics of WSI/WSD are debated in terms of quality [7]. We consider supervised recall and unsupervised F-measure, as in the competition original paper [1]. The first one maps the output of a system to the true senses of the target words' instances and the second one measures the quality of the correspondence between the automatically found clusters and the senses. We consider that the number of senses found by the system is also a rather good indicator of performance: the best competition baseline assigns the most frequent sense to each target word (this baseline is called MFS), thus this baseline system would have an average of 1 sense (cluster) per word. A system that goes near this average may be indeed not resolving the task efficiently but finding the MFS trivial solution. Consequently, to show that we do not fall in the MFS solution, we display in our results the average number of clusters.

Results Word sense induction and disambiguation results are found in Table 5. Again, we aim to surpass the baseline of the single features and early fusion. We experimentally set $\beta = 0.90$ and $\gamma = 50$. In this task, in late fusion, when the first matrix is deemed more relevant than the second one, the performance is higher. This may be due to the fact that, in this task, the feature matrices rows contain types (that is, each line represent an unique word), and thus they are more dense, which may entail more noisy data. By reducing the relevance of the second matrix in late fusion, we are effectively attenuating the less important information. Regarding $\gamma = 50$, again due to the denser characteristic of the matrices, there is a larger quantity of true similar words that are useful to project information into another matrix, through cross fusion.

The WSI/WSD results are shown in Table 5. In the following paragraph, we will discuss these result all at once. Due to the page limit constraint, we omit certain configurations that do not yield interesting results either by converging to the MFS solution (1

Table 5. Supervised Recall and Unsupervised F-measure for the Semeval-2007 corpus. We also display the average number of clusters found by each fusion configuration.

Method	**Recall (%)**			**FM (%)**			**# cl**
	all	noun	verb	all	noun	verb	
							Single Features
M^{LEX}	79.20	82.10	75.80	72.70	76.90	67.90	4.13
M^{SYN}	79.10	81.60	76.20	69.30	69.40	69.20	4.47
							Early Fusion
$E(M^{LEX}, M^{SYN})$	78.70	81.11	76.10	74.00	76.66	71.11	4.46
							Cross Early Fusion
$X(S^{LEX}, M^{LEX})$	79.20	82.30	75.70	76.20	79.60	72.50	3.63
$X(S^{LEX}, M^{SYN})$	78.30	80.90	75.30	74.60	75.10	73.90	3.08
$X(S^{SYN}, M^{LEX})$	78.60	80.90	76.10	78.90	80.70	76.90	1.08
$X(S^{SYN}, M^{SYN})$	78.90	81.40	76.10	73.70	77.70	70.00	2.72
							Cross Late Fusion
$X(S^{SYN}, S^{LEX})$	78.70	80.90	76.20	78.90	80.80	76.80	1.01
$X(S^{LEX}, S^{SYN})$	78.80	80.90	76.06	78.70	80.50	76.80	1.33
							Cross Late Early Fusion
$X(X(S^{LEX}, S^{SYN}), M^{LEX})$	78.40	80.40	76.10	70.00	68.70	71.40	3.11
$X(X(S^{LEX}, S^{SYN}), M^{SYN})$	78.90	81.80	75.60	75.20	77.40	72.80	3.16
							Early Cross Early Fusion
$E(M^{LEX}, X(S^{LEX}, M^{LEX}))$	79.20	82.40	75.70	76.00	79.50	72.10	3.57
$E(M^{SYN}, X(S^{LEX}, M^{LEX}))$	78.30	80.50	75.80	75.20	75.40	75.00	1.95
							Late Cross Early Fusion
$L(M^{SYN}, X(S^{LEX}, M^{SYN}))$	78.60	81.10	75.80	67.80	71.40	63.80	4.22
$L(M^{LEX}, X(S^{LEX}, M^{LEX}))$	79.50	82.80	75.70	76.09	79.10	72.70	3.96
							Early Late Cross Early Fusion
$E(M^{LEX}, L(M^{SYN}, X(S^{LEX}, M^{SYN})))$	78.50	81.40	75.40	74.20	78.20	69.80	4.26
$E(M^{LEX}, L(M^{LEX}, X(S^{LEX}, M^{LEX})))$	79.50	82.70	75.90	75.80	78.50	72.70	3.99

sense found per target word) or because the performance shown by those configurations is not interesting.

Regarding Single Features, M^{LEX} comes on top of M^{SYN} again. Nonetheless, M^{SYN} is much closer in terms of performance, and as expected, it is actually higher with regards to verbs.

On the 1F level, we see that the early fusion technique in this task does not surpass the independent features representation. Our intuition is that the similarities of both matrices seem to be correlated. In cross early fusion, the best result is obtained by

$X(S^{LEX}, M^{LEX})$, regarding the unsupervised F-measure. This configuration already beats our baselines, improving both noun and verb results on the unsupervised evaluation, improving the supervised recall of nouns, and staying on the same level considering all words. Also, it produces more senses than the MSF average number of senses (1 sense per target word), which is good but not indicative of results correctness. Regarding cross late fusion, given the average number of clusters produced, it seems that both results converge towards the MFS, therefore we do not consider these results.

Beginning with the fusion recombinations, in level 2F, both cross late early fusions yield average results. In cross early cross early fusion, the early fusion of M^{LEX} with $X(S^{LEX}, M^{LEX})$ yields very similar results than $X(S^{LEX}, M^{LEX})$. The next natural step is to test this fusion via a linear combination, with a late fusion. The result obtained confirmed the intuition of enriching a single feature matrix with another weighted-down matrix to improve the performance. Indeed, we consider that $L(M^{LEX}, X(S^{LEX}, M^{LEX}))$ gets the best results in terms of all-words supervised recall and the second best all-words unsupervised F-measure (we do not consider solutions that are too close to the MFS baseline).

We test the same configurations as in NER, within the NF level, to try and improve our results. Nonetheless, in general, they do not overcome the best result found previously.

In general, we found that the recombination fusion techniques work in terms of improving the performance of the tasks addressed. In the following, we make our final remarks and the future work to be done regarding fusion techniques on NLP tasks.

5 Conclusion and Future Work

In this paper, we presented a comparative study of multimedia fusion techniques applied to two NLP tasks: Named Entity Recognition and Word Sense Induction and Disambiguation. We also proposed new fusion recombinations in order to complement the information contained in the single representation matrices. In order to accomplish this goal, we built upon basic fusion techniques such as early and late fusion, as well as cross media fusion to transfer quality information from one set of features to another.

We found that by taking a strong feature, in our case lexical context, M^{LEX}, and enriching it with the output of rather complex fusion combinations, we can improve the performance of the tasks addressed. The enrichment has to give more relevance to the strong feature matrix, by selecting the right parameters.

While there is an improvement, we do note that fusion techniques augment the computing time and memory consumption of the tasks at hand by enlarging the feature space or by making it more dense. In that sense, more intelligent ways of finding the most appropriate fusion must be researched. This is indeed one of our future work paths: determining an optimal fusion path from single features to a N-degree fusion recombination. Coupled with this, the automatic determination of the parameters is still ongoing research in the multimedia fusion community. Consequently, we believe that efficiently determining both parameters and fusion combinations is the general domain of our future work. Another route we would like to explore is testing these techniques on other tasks and with datasets from different domains, in order to assert its effectiveness.

References

1. Agirre, E., Soroa, A.: Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In: Proceedings of the 4th International Workshop on Semantic Evaluations. pp. 7–12. SemEval '07, Association for Computational Linguistics, Stroudsburg, PA, USA (2007)
2. Ah-Pine, J., Csurka, G., Clinchant, S.: Unsupervised visual and textual information fusion in CBMIR using graph-based methods. ACM Trans. Inf. Syst. 33(2), 9:1–9:31 (2015)
3. Atrey, P.K., Hossain, M.A., El-Saddik, A., Kankanhalli, M.S.: Multimodal fusion for multimedia analysis: a survey. Multimedia Syst. 16(6), 345–379 (2010)
4. Balasuriya, D., Ringland, N., Nothman, J., Murphy, T., Curran, J.R.: Named entity recognition in wikipedia. In: Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources. pp. 10–18. People's Web '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)
5. Clinchant, S., Ah-Pine, J., Csurka, G.: Semantic combination of textual and visual information in multimedia retrieval. In: ICMR. p. 44. ACM (2011)
6. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10. pp. 1–8. EMNLP '02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002)
7. de Cruys, T.V., Apidianaki, M.: Latent semantic word sense induction and disambiguation. In: ACL. pp. 1476–1485. The Association for Computer Linguistics (2011)
8. Daume, III, H.C.: Practical Structured Learning Techniques for Natural Language Processing. Ph.D. thesis, Los Angeles, CA, USA (2006), aAI3337548
9. Gialampoukidis, I., Moumtzidou, A., Liparas, D., Vrochidis, S., Kompatsiaris, I.: A hybrid graph-based and non-linear late fusion approach for multimedia retrieval. In: CBMI. pp. 1–6. IEEE (2016)
10. Goyal, K., Hovy, E.H.: Unsupervised word sense induction using distributional statistics. In: COLING. pp. 1302–1310. ACL (2014)
11. Levy, O., Goldberg, Y.: Dependency-based word embeddings. In: ACL (2). pp. 302–308. The Association for Computer Linguistics (2014)
12. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations. pp. 55–60 (2014)
13. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes 30(1), 3–26 (2007)
14. Navigli, R.: Word sense disambiguation: A survey. ACM Computing Surveys (CSUR) 41(2), 10 (2009)
15. Panchenko, A., Faralli, S., Ponzetto, S.P., Biemann, C.: Unsupervised does not mean uninterpretable: The case for word sense induction and disambiguation. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017). The Association for Computer Linguistics (2017)
16. Ratinov, L., Roth, D.: Design challenges and misconceptions in named entity recognition. In: CoNLL. pp. 147–155. ACL (2009)
17. Sang, E.F.T.K., Meulder, F.D.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: CoNLL. pp. 142–147. ACL (2003)
18. Yu, S.I., Jiang, L., Mao, Z., Chang, X., Du, X., Gan, C., Lan, Z., Xu, Z., Li, X., Cai, Y., et al.: Informedia@ trecvid 2014 med and mer. In: NIST TRECVID Video Retrieval Evaluation Workshop. vol. 24 (2014)