Data Augmentation for Hypernymy Detection

Thomas Kober[★], **Julie Weeds**[♠], **Lorenzo Bertolini**[♠] and **David Weir**[♠]
[★]rasa
Schönhauser Allee 175, 10119 Berlin
t.kober@rasa.com

[♠]University of Sussex
Brighton, BN1 9RH
{j.e.weeds, l.bertolini, d.j.weir}@sussex.ac.uk

Abstract

The automatic detection of hypernymy relationships represents a challenging problem in NLP. The successful application of state-of-the-art supervised approaches using distributed representations has generally been impeded by the limited availability of high quality training data. We have developed two novel data augmentation techniques which generate new training examples from existing ones. First, we combine the linguistic principles of hypernym transitivity and intersective modifier-noun composition to generate additional pairs of vectors, such as *small dog - dog* or *small dog - animal*, for which a hypernymy relationship can be assumed. Second, we use generative adversarial networks (GANs) to generate pairs of vectors for which the hypernymy relation can also be assumed. We furthermore present two complementary strategies for *extending* an existing dataset by leveraging linguistic resources such as WordNet. Using an evaluation across 3 different datasets for hypernymy detection and 2 different vector spaces, we demonstrate that both of the proposed automatic data augmentation and dataset extension strategies substantially improve classifier performance.

1 Introduction

The detection of hypernymy relationships between terms represents a challenging commonsense inference problem and is a major component in recognising paraphrase and textual entailment in larger units of text. Consequently, it has important applications in Question-Answering, Text Simplification and Automatic Summarization. For example,

There are lots of cars and vans at the port today.

might be adequately summarised by

There are lots of vehicles at the port today.

because *car* and *van* both lexically entail, i.e. they are both hyponyms of the more general term *vehicle*.

While distributed representations of words are commonly used to find semantically similar words, they do not straightforwardly provide a way to distinguish more fine-grained semantic information, such as hypernymy, co-hyponymy and meronymy relationships. This deficiency has attracted substantial attention in the literature, and with regard to the task of hypernymy detection, both unsupervised approaches (Hearst, 1992; Weeds et al., 2004; Kotlerman et al., 2010; Santus et al., 2014; Rimell, 2014; Nguyen et al., 2017; Chang et al., 2018) and supervised approaches (Weeds et al., 2014; Roller et al., 2014; Roller and Erk, 2016; Shwartz et al., 2016; Vulić and Mrkšić, 2018; Rei et al., 2018) have been proposed.

Supervised methods have, however, been severely hampered by a lack of adequate training data. Not only has a paucity of labelled data been an obstacle in the adoption of deep neural networks and other more complex supervised methods, but two compounding problem-specific issues have been identified. First, there is a need to avoid lexical overlap between the training and test sets in order to avoid the lexical memorisation problem (Weeds et al., 2014; Levy et al., 2015a), where a supervised method simply learns the relationships between lexemes rather than generalising to their distributional features. Second, the performance of classifiers given just the hypernym word (at training and testing) has been shown to be almost as good as performance given both words (Weeds et al., 2014; Shwartz et al., 2017). This suggests that classifiers are learning the distributional features that make something a more general term or a more specific term. Our conjecture is that in order to learn the more complex function, more complex machinery is required, and hence more labelled data is required.

In computer vision and other fields where machine learning is applied, it is common place to use data augmentation techniques to increase the size of the training data (Shrivastava et al., 2017; Park et al., 2019). The idea is that there are certain transformations of the data under which the class label remains invariant. For example, rotating an image does not have an impact on whether that image contains a face or not. By providing a supervised classifier with rotated examples, it can better generalise.

In this paper, we consider the use of linguistic transformations to augment existing small datasets for hypernymy detection. The challenge is to identify transformations that can be applied to the representations of two words that are known to be in a hypernym relationship, such that the entailment relationship still holds between the transformed representations. We have two proposals as to how this can be achieved.

Our first augmentation technique is based on the hypothesis that lexical entailment is transitive and therefore invariant under certain compositions. For example, if A entails B and B entails C then A also entails C. Suitable candidates for A can be found by composing common intersective adjectives with the noun B. For example, if we know that *car* entails *vehicle*, then we can augment the dataset with *fast car* entails *car* and *fast car* entails *vehicle*.

Our second augmentation technique is based on the hypothesis that lexical entailment is invariant within a certain threshold of similarity. If A entails B, A' is very similar to A and B' is very similar to B then A' will also entail B'. In order to obtain vectors which are sufficiently similar to the words in the training data, we apply generative adversarial networks (GANs) to create realistic-looking synthetic vectors, from which we choose the most similar to the words in the training data.

We evaluate the proposed techniques on three hypernymy detection datasets. The first two are standard benchmark tasks in this area (Weeds et al., 2014; Baroni et al., 2012), both of which are generated from WordNet (Fellbaum, 1998). However, since many of the approaches to hypernmy classification involve vector space models which have been specialised using the entirety of WordNet, we need to guard against the danger that evaluations are simply measuring how well WordNet has been encoded, rather than how well the general hypernymy relationship has been learned. In light of this, we introduce a new dataset for evaluation (that we call **HP4K**) which does not rely on WordNet in its construction.

We evaluate the performance of our two data augmentation techniques against two methods for increasing the size of the training data which rely on finding or mining more non-synthetic examples. First, we consider the extraction of additional examples from WordNet. Second, we consider extracting examples automatically from a Wikipedia corpus using Hearst Patterns (Hearst, 1992). This provides us with what one would expect to be an upper bound on what we might reasonably expect to achieve with a similar amount of synthetic examples generated using our data augmentation techniques.

Our contributions are thus threefold. First, we have identified two novel data augmentation techniques for the task of hypernymy detection which have the potential to generate almost limitless quantities of synthetic data. Second, we show, rather surprisingly, that adding synthetic data is more effective than adding non-synthetic data in almost all cases. Third, we release a new benchmark evaluation dataset for the lexical entailment task that is not dependent on WordNet.

2 Related Work

While to the best of our knowledge this work represents the first approach to lexical entailment (LE) via data augmentation, a number of alternative approaches have been proposed. Many works have focused on supervised methods for injecting an external source of knowledge into distributional representations. Introduced by the *retro-fitting* algorithm (Faruqui et al., 2015), vector-specialization methods modify existing representations to embed desired features. Recently, Vulić and Mrkšić (2018) have proposed LEAR to specifically solve lexical entailment by encoding a hierarchical knowledge in the vectors' norm. Similar supervised techniques have also involved neural network architectures, such as SDSN Rei et al. (2018) that learns transformation functions in order to generate task specific embeddings from task-agnostic ones.

Other authors have proposed models to embed LE relations without external knowledge, in an unsupervised fashion. Nickel and Kiela (2017), for example, introduced an *n*-dimensional Poincaré ball based model to generate representations that embed a hierarchical relation between words in addition to a similarity-based one. Other methods, such as the model of Henderson and Popa (2016), which represents an expansion of word2vec based on mean-field approximation, generates a distributional space focused on LE alone.

Tasks involving entailment relations above the phrase level, such as natural language inference (NLI) and textual entailment, have been primarily approached with rich neural network architectures, which have also been shown to benefit from the injection of external knowledge (Chen et al., 2018). Notably, Kang et al. (2018) leverage lexical resources such as WordNet and apply a similar GAN-based model for data augmentation for textual entailment.

The use of data augmentation techniques is still limited within the field of NLP, where it has primarily been used for machine translation (Sennrich et al., 2016; Fadaee et al., 2017; Wang et al., 2018).

3 Data Augmentation Strategies for Hypernymy Detection

Given a labelled dataset $\mathcal{D}_{\mathcal{X}}$ of triples $\langle x_{hypo}^{(i)}, x_{hyper}^{(i)}, y^{(i)} \rangle$, where $x_{hypo}^{(i)}, x_{hyper}^{(i)} \in \mathcal{X}$ and $y^{(i)} \in \{0, 1\}$, we define data augmentation as adding additional hyponym-hypernym triples $\langle x'^{(j)}_{hypo}, x'^{(j)}_{hyper}, y'^{(j)} \rangle$ coming from an automatically generated augmentation set $\mathcal{A}_{\mathcal{X}'}$, such that $x'^{(j)}_{hypo}, x'^{(j)}_{hyper} \in \mathcal{X}'$ and $y'^{(j)} \in \{0, 1\}$, to the existing training set of $\mathcal{D}$. We ensure that the data augmentation does not introduce any lexical overlap with the existing test set, i.e. $\mathcal{X} \cap \mathcal{X}' = \emptyset$.

We distinguish between *data augmentation* and *dataset extension*, where in the former case we only leverage knowledge from the existing dataset and in the latter case we rely on expanding the training set with additionally mined hyponym-hypernym pairs. Below, we discuss two ways of augmenting and two ways of extending a training set. We make use of a cleaned October 2013 Wikipedia dump (Wilson, 2015) as reference corpus to determine word and bigram frequencies.

Distributional Composition based Augmentation. Here, we take a modified noun as being in a hypernymy relation with the unmodified noun. For example, we treat the pairs ⟨*fast car*, *car*⟩ and ⟨*car*, *vehicle*⟩ as expressing the same semantic relation when the modifier-noun compound is composed with an intersective composition function.

We focus on adjective-noun (AN) and noun-noun (NN) compounds, extracted from our reference corpus where each AN or NN compound occurred at least 50 times. We filtered out pairs that included non-subsective adjectives using a wordlist from Nayak et al. (2014)[1].

We consider two strategies for automatically constructing positive hyponym-hypernym pairs: simple positive cases such as ⟨*small dog*, *dog*⟩ or ⟨*fast car*, *car*⟩; and gapped positive cases that mimic the transitivity of hypernym relations, where we pair the hypernym of an existing hyponym-hypernym pair with a compound hyponym. For example if ⟨*car*, *vehicle*⟩ is in the training data, we combine *car* with one of its modifiers to create the pair ⟨*fast car*, *vehicle*⟩.

We construct negative pairs from the simple positive cases using two strategies: creating compositional co-hyponyms such as ⟨*fast car*, *red car*⟩, where we keep the head noun fixed and pair it with two different modifiers; and creating perturbed simple positive examples, such as ⟨*small dog*, *cat*⟩ where we select the incorrect hypernym (e.g. *cat*) from the n most similar nouns to the composed hyponym (e.g. *dog*). We apply the same methodology to the perturbed gapped positive examples, replacing the correct hypernym with a noun from the top n neighbours of the compositional hyponym's head noun. For example, given a positive pair such as ⟨*dog*, *animal*⟩, this would result in negative examples such as ⟨*small dog*, *vehicle*⟩, where the hyponym *dog* is paired with a modifier and the hypernym *animal* is replaced with one of its neighbours, in this case, *vehicle*.

In neural word embeddings, an additive composition function approximates the intersection of the corresponding feature spaces (Tian et al., 2017), hence by creating positive pairs such as ⟨*small dog*, *dog*⟩,

[1] In preliminary experiments we did not find that filtering non-subsective adjectives had much of an effect, but decided to move forward with the filtered data nonetheless.

we encode the distributional inclusion hypothesis (Weeds et al., 2004; Geffet and Dagan, 2005) in the augmentation set.

GAN based Augmentation. We create an augmentation set using Generative Adversarial Networks (Goodfellow et al., 2014). GANs consist of two model components — the generator and the discriminator — which are typically implemented as neural networks. The generator's task is to create data that mimics the distribution of the original data, while the discriminator's task is to distinguish between data coming from the real distribution and synthetic data coming from the generator. Both components are trained jointly until the generator succeeds in creating realistic data. Using GANs for data augmentation has been shown to be a successful strategy for a number of computer vision tasks (Shrivastava et al., 2017; Frid-Adar et al., 2018; Neff, 2018). Our goal is to create synthetic hyponym-hypernym pairs that are similar to real examples. Unlike most other scenarios involving GANs for NLP tasks, our generated vectors do not need to correspond to actual words.

For our model - *GANDALF*[2] - we used a list of about 40K noun lemmas for which we had vector representations as the 'real' data input to *GANDALF* and sampled the synthetic vectors from a Gaussian distribution, optimising a binary cross-entropy error criterion for the generator and the discriminator, which are both simple feedforward networks with a single hidden layer. We provide *GANDALF*'s full model details in Appendix A. As an additional quality check for the generated vectors, we tested whether a logistic regression classifier could distinguish the synthetic and non-synthetic vectors. Typically, the accuracy of the linear classifier was between 0.55-0.65 which we considered as sufficient for our purposes[3].

Once *GANDALF* has been trained, the generator is used to create a large collection[4] of synthetic noun vectors. To augment a dataset, $\mathcal{D}_{\mathcal{X}}$, for each triple, $\langle x_{hypo}, x_{hyper}, y\rangle \in \mathcal{D}_{\mathcal{X}}$ we find the n synthetic vectors most similar to x_{hypo} and the n synthetic vectors most similar to x_{hyper} and for each of the n^2 synthetic vector pairs, $\langle x'_{hypo}, x'_{hyper}\rangle$, we create the triple $\langle x'_{hypo}, x'_{hyper}, y\rangle$. The augmented training set is formed by randomly sub-sampling this set of triples.

WordNet based Extension. WordNet (Fellbaum, 1998) is a manually curated large-scale lexical resource, covering a wide range of different lexical relations between words, where groups of semantically similar words form 'synsets'[5]. For each synset we extract all hypernyms and hyponyms of a given lexeme, and add it as a positive hyponym-hypernym pair if the original lexeme and any extracted hypernym/hyponym occurs at least 30 times in our reference corpus.

We construct negative augmentation pairs based on distributional similarity, where we calculate the pairwise cosine similarities between all lexemes in the positive set. Subsequently we use all antecedent (LHS) lexemes from the extracted positive pairs and select the top n most similar words for each antecedent as negative examples[6]. We also tried selecting the top n least similar words for each antecedent as negative examples and randomly shuffling the positive pairs, but did not observe any significant differences in performance.

Pattern based Extension. Hearst Patterns (Hearst, 1992) are textual patterns such as *a car is a vehicle* and can be automatically mined from text corpora in an unsupervised way. This has recently been shown to deliver strong performance on the hypernymy detection task (Roller et al., 2018). In this work, we leverage Hearst Patterns to mine additional hyponym-hypernym pairs in order to extend a training set. We treat any extracted noun pairs as additional positive examples and create the negative pairs in the same way as for the WordNet-based approach above.

[2] **GAN**-based **D**ata **A**ugmentation for **L**exical in**F**erence.

[3] Augmenting the loss of the generator with the performance of the auxiliary linear classifier did not turn out to be beneficial.

[4] We would typically create half a million synthetic nouns.

[5] We use the API provided by NLTK (Loper and Bird, 2002), using WordNet 3.0.

[6] Ensuring we don't accidentally include any real positive pairs.

4 Experiments

4.1 Datasets

We evaluate models on the datasets **Weeds** (Weeds et al., 2014) and **LEDS** (Baroni et al., 2012): well-studied and frequently used benchmarks for the hypernymy detection task (Roller et al., 2014; Vilnis and McCallum, 2014; Roller and Erk, 2016; Carmona and Riedel, 2017; Shwartz et al., 2017). Since both datasets use WordNet during the construction process, this can give rise to a bias in favour of those models that also make use of WordNet. To address this concern, we have created a new entailment dataset, **HP4K**, that makes use of Hearst Patterns, and is manually annotated, thereby avoiding the use of WordNet.

Weeds: The dataset is based on nouns sampled from WordNet where each noun had to occur at least 100 times in Wikipedia, and its predominant sense had to account for more than 50% of the occurrences in SemCor (Miller et al., 1993). We use the predefined split of Weeds et al. (2014), that avoids any lexical overlap between the training and evaluation sets. The split contains 2012 examples in the training set, evenly balanced between positive and negative hyponym-hypernym pairs, and 502 examples in the evaluation set.

LEDS: The dataset consists of 2770 examples, evenly balanced between positive and negative hyponym-hypernym pairs. The positive examples are based on direct hyponym-hypernym relationships from WordNet and the negative examples are based on a random permutation of the hypernyms of the positive pairs. As there is no predefined training/evaluation split, we make use of the 20-fold cross-validation methodology of Roller and Erk (2016) that avoids any lexical overlap between training and evaluation sets.

HP4K: We extracted Hearst Patterns from our reference Wikipedia corpus and randomly selected 4500 unigram pairs. Subsequently, we manually annotated each pair according to whether it constitutes a correct hyponymy-hypernymy relation or not. The annotation was carried out by 4 people — two native speakers and two non-native speakers of English. We built two teams, each consisting of a native and a non-native speaker of English. Each team then annotated one half of the dataset. The initial round of annotations resulted in a Cohen's κ score of 0.714, indicating substantial agreement (Viera and Garrett, 2005). Conflicts were resolved by the two native speakers, such that the native speaker of team A would resolve team B's annotation conflicts and vice-versa.

During annotation we noticed that positive pairs typically fall into one of two categories. Either they were 'true' subtype-supertype relations, such as $\langle dog, animal \rangle$, or they were individual-class relationships where the hyponym is typically a named entity and represents a *specific* instance of the more general class, as for example in $\langle Nirvana, band \rangle$. Negative pairs were of a more diverse nature and included a range of different relations, such as co-hyponyms, meronyms or reverse hyponym-hypernyms. Negative pairs can also be comprised of two random nouns or two nouns without any semantic relation due to some amount of noise[7] in extracting candidates solely on the basis of Hearst Patterns. Table 1 shows positive and negative examples from the dataset.

Pair	**Relationship**	**Label**
$\langle dog, animal \rangle$	hyponymy-hypernymy (Subtype)	*True*
$\langle Nirvana, band \rangle$	hyponymy-hypernymy (Individual)	*True*
$\langle beef, stew \rangle$	meronymy	*False*
$\langle pedestrian, road \rangle$	topical relatedness	*False*
$\langle chemical, adenosine \rangle$	reverse hyponymy-hypernymy	*False*
$\langle cherry, plum \rangle$	co-hyponymy	*False*
$\langle form, situation \rangle$	none	*False*

Table 1: Positive and Negative examples from our proposed **HP4K** dataset.

The final dataset consists of 4369 pairs with a class distribution of 45 : 55 (positive : negative). Subsequently we split the dataset into a training and evaluation split and ensured that there is no lexical overlap between the two sets, resulting in a training set of size 3426 and an evaluation set of size 943[8].

[7] We discarded examples where neither of the two words was a noun, e.g. pairs such as $\langle you, who \rangle$.

[8] All resources are available from `https://github.com/tttthomasssss/le-augmentation`.

4.2 Models

We conduct experiments with two distributional vector space models, word2vec (Mikolov et al., 2013) and HyperVec (Nguyen et al., 2017). HyperVec is based on word2vec's skip-gram architecture and leverages WordNet to optimise the word representations for the hypernym detection task. Hierarchical information is encoded in the norm of the learned vectors, such that lexemes higher up in the hypernymy hierarchy have larger norms than lexemes in lower parts.

For word2vec we use the 300-dimensional pre-trained Google News vectors[9] and for HyperVec we trained 100-dimensional embeddings on the cleaned October 2013 Wikipedia dump (Wilson, 2015) using the recommended settings of Nguyen et al. (2017), as our augmentation sets contained a large proportion of words that were OOV in the pre-trained HyperVec vectors[10].

In our experiments, we consider a supervised scenario where a classifier predicts a hyponym-hypernym relation between two given word embeddings. We use two different models as classifier a logistic regression classifier (LR), and a 3-layer feedforward neural network (FF). In both cases, the classifier takes the aggregated hypothesised hyponym-hypernym pair as input and predicts whether the pair is in a hyponym-hypernym relation. We report a detailed overview of the model parameterisation in Appendix A.

The two models share the same procedure for aggregating the word embeddings of the hypothesised hyponym-hypernym pair. For data augmentation based on distributional composition, we use vector averaging as composition function, which we found gave substantially better performance than addition in preliminary experiments.

4.3 Results

For the FF network, we performed 10-fold cross-validation on the **Weeds** and **HP4K** training sets. As our evaluation for **LEDS** is based on a 20-fold cross-valiation split, rather than a pre-defined training/evaluation split as for **Weeds** and **HP4K**, the same procedure for hyperparameter tuning is not straightforwardly applicable without exposing the model to some of the evaluation data. However, we found that the top parameterisations for **Weeds** and **HP4K** were quite similar and therefore applied a parameterisation to the FF model for **LEDS** that performed well in 10-fold cross-validation on **Weeds** *and* **HP4K**. For data augmentation and dataset extension, we consider the following amounts of additional data: $\{200, 1000, 2000, 4000, 10000, 20000, 40000\}$. All augmentation sets are balanced between positive and negative pairs.

Figure 1 shows the increase in absolute points of accuracy for the LR and FF model, as well as both vector spaces, averaged across all datasets. While in total data augmentation as well as dataset extension has a positive impact, the gains are larger for the FF model, suggesting that a higher capacity model is necessary to more effectively leverage the additional information from the augmentation source. Furthermore, before starting our experiments we exptected that extending an existing dataset with WordNet represents an upper bound on performance, given that WordNet is a large repository of human annotated and curated data. However in our experiments we found that data augmentation by either distributional composition or by using *GANDALF* remarkably *surpassed* performance of the WordNet-based extension technique regularly.

The effect of data augmentation and dataset extension in absolute points of accuracy on each dataset individually for the FF model is shown in Figure 2. It highlights consistent improvements across the board with only a single performance degradation in the case of extending the LEDS dataset with Hearst Patterns when using HyperVec-based word representations. The results per dataset for the LR model are presented in Appendix A and show that the LR model is less effective in leveraging the augmented data, causing more frequent performance drops. This suggests that models with more capacity are able to make more efficient use of additional data and are more robust in the presence of noise which is inevitably introduced by automatic methods.

Table 2 compares our FF model using word2vec embeddings with all proposed techniques for augmenting or extending a dataset. Our techniques are able to outperform a non-augmented model by 4-6

[9] Available from: `https://code.google.com/archive/p/word2vec/`.

[10] We used the HyperVec code from `www.ims.uni-stuttgart.de/data/hypervec`.

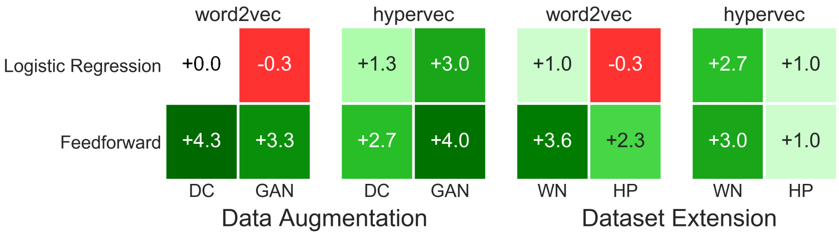

Figure 1: Effect of data augmentation and dataset extension in absolute points of accuracy averaged across all datasets over the same model without augmentation or extension. The 2 heatmaps on the left are based on data augmentation (DC=Distributional Composition, GAN=*GANDALF*) and the 2 heatmaps on the right are based on dataset extension (WN=WordNet, HP=Hearst Patterns).

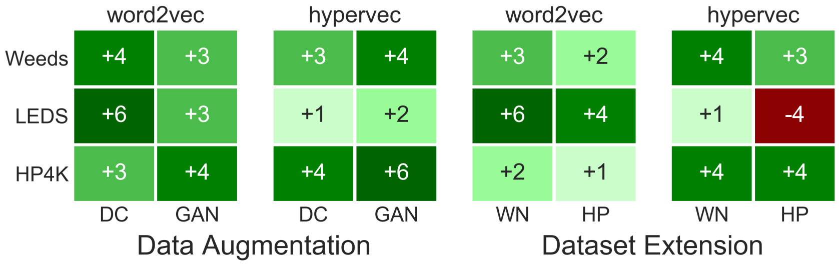

Figure 2: Effect of data augmentation and dataset extension in absolute points of accuracy on all datasets for the FF model.

points in accuracy, representing a relative error reduction of 14%-26%. While the primary objective in this work is to improve an existing model setup with data augmentation, our augmented models compare favourably with previously published results.[11]. In general, data augmentation by distributional composition or by *GANDALF* overcomes two key weaknesses of simply extending a dataset with more data from WordNet or Hearst Patterns. First, many of the hyponym-hypernym pairs we mined from WordNet contain low-frequency words, which may have poor representations in our vector space models. Second, while using Hearst Patterns typically returned higher frequency words, the retrieved candidates frequently did not represent hyponymy-hypernymy relationships.

Model	**Weeds**	**LEDS**	**HP4K**
Baseline - No Augmentation/Extension	0.72	0.77	0.67
Distributional Composition Augmentation	**0.76**	**0.83**	0.70
GANDALF Augmentation	0.75	0.80	**0.71**
WordNet Extension	0.75	**0.83**	0.69
Hearst Patterns Extension	0.74	0.81	0.68
Weeds et al.(2014)	0.75	-	-
Carmona and Riedel (2017)	0.63	0.81	-

Table 2: Accuracy scores for the data augmentation and the two dataset extension strategies in comparison to the same FF model without any augmentation or extension.

5 Analysis

The concrete amount of data augmentation, i.e. the number of additional hyponym-hypernym pairs that are added to the training set, represents a tuneable parameter. Figure 3 shows the effect of varying amounts of data augmentation for the FF model, using word2vec representations, across all datasets. We note that all amounts of additional augmentation data share the same quality, i.e. it is not the case that a smaller augmentation set consists of "better data" or contains less noise than a larger set. For the **Weeds** and **LEDS** datasets, peak performance is typically achieved with smaller amounts of additional data, whereas for the **HP4K** dataset optimal performance is achieved with larger amounts of augmentation

[11] We note that due to the use of different performance metrics and cross-validation splits, direct model-to-model comparisons are difficult on the LEDS and Weeds datasets. Thus we only compare to approaches that use the same evaluation protocol as we do.

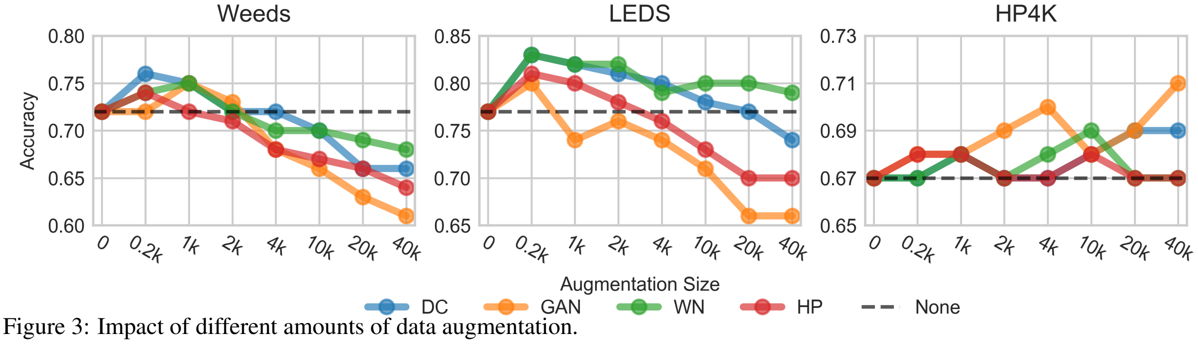

Figure 3: Impact of different amounts of data augmentation.

data. One explanation for the different augmentation characteristics of the **HP4K** dataset in comparison to the other two datasets is its independence of WordNet during the development of the dataset.

5.1 Data Augmentation in Space

In order to visualise what area of the vector space the *GANDALF* vectors and the composed vectors inhabit, we create a t-SNE (van der Maaten and Hinton, 2008) projection of the vector spaces in Figure 4. For the visualisation we produced the nearest neighbours of standard word2vec embeddings and augmentation embeddings for 5 exemplary words and project them into the same space. Figure 4 shows that the generated augmentation points, marked with an "x", fit in with the real neighbours and do not deviate much from the "natural" neighbourhood of a given word. *GANDALF* vectors typically inhabit the edges of a neighbour cluster, whereas composed vectors are frequently closer to the cluster centroid. Table 3 lists the nearest neighbours for the example words. For word2vec and the composed augmentation vectors, we simply list the nearest neighbours of each query word. For *GANDALF* we list the nearest neighbours of the generated vector that correspond to actual words. For example, if the vector *GANDALF-234* is closest to the representations of *sugar*, *GANDALF-451* and *mountain*, we only list *sugar* and *mountain* as neighbours of *GANDALF-234*.

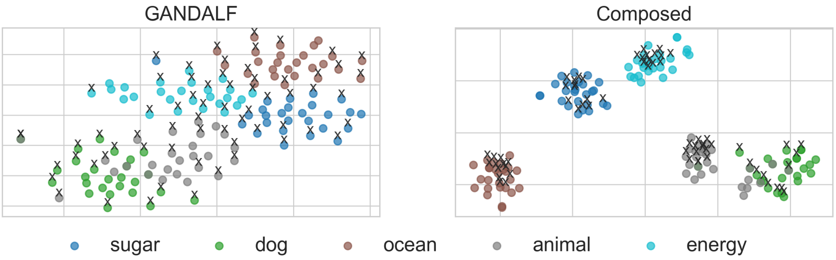

Figure 4: t-SNE visualisation of the data augmentation spaces. Data points marked with "x" denote the representation as coming from the data augmentation set.

Word	word2vec Neighbours	*GANDALF* Neighbours	Composed Neighbours
sugar	refined sugar, cane sugar, turbinado, cocoa, sugars	cmh rawalpindi, prescribed antipsychotic medication, sugar, mumtaz bhutto, akpeteshie	raw sugar, white sugar, brown sugar, sugar price, sugar industry
dog	dogs, puppy, pit bull, pooch, cat	ellis burks, sniffing glue, microchip implants, liz klekamp, cf rocco baldelli	pet cat, dog fighting, cat breed, rat terrier terrier breed
ocean	sea, oceans, pacific ocean, atlantic ocean, oceanic	pacific ocean, heavily vegetated, alaska aleutian, seagrasses, plutoid	ocean basin, pacific ocean, shallow sea, sea fish, sea mammal
animal	animals, animal welfare, dog, pet, cats	animal, pet, hallway feeds, poop scooping, panhandle animal welfare	first animal, adult animal, zoo animal, animal, animal organization, different
energy	renewable energy, enery, electricity, enegy, fossil fuel	radial velocity measurements, stopped, renewable energy, bicycle advisory, steinkuehler	energy efficiency, solar energy, state energy, food energy, energy company

Table 3: Nearest neighbours for word2vec, GAN vectors and composed vectors.

The composed neighbours for each word are typically closely related to the original query, e.g. *raw sugar* for *sugar*, or *zoo animal* for *animal*. The *GANDALF* neighbours on the other hand have a much weaker association with the query word, but are frequently still related to it *somehow* as in the example of *akpeteshie*, which is a spirit on sugar cane basis, as a neighbour for *sugar*.

5.2 Data Augmentation as Regularisation

In the past, a prominent criticism of distributional methods for hypernymy detection was that such models were found to frequently identify features of a prototypical hypernym in the distributional representations, rather than being able to dynamically focus on the relevant features that are indicative of a hypernymy relation for a specific pair of words (Weeds et al., 2014; Levy et al., 2015b). We therefore briefly investigate whether data augmentation can be used as a regularisation mechanism that helps prevent models from overfitting on prototypical hypernym features.

Table 4 shows the results on the **Weeds** dataset using a hypernym-only FF model with word2vec representations, in comparison to the same model variant that makes use of the hyponym and the hypernym. Ideally, we would hope to see weak performance for the hypernym-only and strong performance on the full model. This would indicate that the classifier does not rely on prototypical features in the hypernym, but is able to focus on specific features in a given hyponym-hypernym pair. For data augmentation by distributional composition there appears to be a correlation between the performance of the hypernym-only and the full model, i.e. a stronger model on the whole dataset also results in better performance for the hypernymy-only model. Hence augmentation by distributional composition might not be effective in helping the model to generalise in its current form. For augmentation with *GANDALF* however, performance for the full model improves, while performance of the hypernym-only model slightly drops, suggesting that the evoked *GANDALF* representations have a regularisation effect, while also improving generalisation. Hence, a fruitful avenue for future work will be further leveraging data augmentation for regularisation.

Augmentation	Hypernym-Only	Full Model
None	0.59	0.72
DC (size=100)	0.60	**0.74**
DC (size=500)	**0.57**	0.71
GAN (size=500)	**0.58**	**0.75**
GAN (size=1000)	0.60	0.73

Table 4: Accuracy for the hypernym-only and full models on the **Weeds** dataset with no, DC or GAN augmentation.

6 Conclusion

It is well established that complex supervised machine learning models require very large quantities of training data. In NLP, in contrast to Computer Vision, data augmentation has not been applied as standard due to the apparent lack of universal rules for label-invariant language transformations.

We have considered the problem of hypernymy detection, and proposed two novel techniques for data augmentation. These techniques rely on semantic rules rather than an external knowledge source, and have the potential to generate almost limitless synthetic data for this task. We demonstrate that these techniques perform as well as, and in most cases better than extending the training set with additional non-synthetic data that is drawn either from an external knowledge source or mined from corpora. Our results are consistent across different evaluation benchmarks, different word vector spaces and different classification architectures. We have thus demonstrated the power of a more complex supervised model combined with an augmented training set at this task. We have also shown that our approach is effective even when the word vector space model has already been specialised for hypernymy detection.

Since WordNet is widely used as a source of information about semantic relations, we have also proposed a new evaluation benchmark which is independent of WordNet. Whilst results are lower across the board on this evaluation, suggesting that it is more difficult than the others, we see the same pattern of increasing performance with a specialised word vector space, a more complex classifier and the use of data augmentation.

Future work includes extending the data augmentation approach to other semantic relations (e.g., co-hyponymy and meronymy) and to other tasks. Development of more complex models for hypernymy detection can now also be carried out which can be trained with augmented datasets. Lastly, we plan to apply our data augmentation approaches in a multilingual setup.

References

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France, April. Association for Computational Linguistics.

Vicente Ivan Sanchez Carmona and Sebastian Riedel. 2017. How well can we predict hypernyms from word embeddings? a dataset-centric analysis. In *Proceedings of the European Chapter of the Association for Computational Linguistics*.

Haw-Shiuan Chang, Ziyun Wang, Luke Vilnis, and Andrew McCallum. 2018. Distributional inclusion vector embedding for unsupervised hypernymy detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 485–495, New Orleans, Louisiana, June. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417, Melbourne, Australia, July. Association for Computational Linguistics.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada, July. Association for Computational Linguistics.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado, May–June. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan. 2018. Synthetic data augmentation using gan for improved liver lesion classification. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 289–293, April.

M. Geffet and I. Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 107–114. Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the International Conference on Computational Linguistics*.

James Henderson and Diana Popa. 2016. A vector space for distributional semantics for entailment. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2052–2062, Berlin, Germany, August. Association for Computational Linguistics.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org.

Dongyeop Kang, Tushar Khot, Ashish Sabharwal, and Eduard Hovy. 2018. AdvEntuRe: Adversarial training for textual entailment with knowledge-guided examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2418–2428, Melbourne, Australia, July. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Special Issue of Natural Language Engineering on Distributional Lexical Semantics*, 4(16):359–389.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015a. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015b. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado, May–June.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, pages 63–70. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Proceedings of the Arpa Workshop on Human Language Technology*, pages 303–308.

Neha Nayak, Mark Kowarsky, Gabor Angeli, and Christopher D. Manning. 2014. A dictionary of nonsubsective adjectives. Technical Report CSTR 2014-04, Department of Computer Science, Stanford University, October.

Thomas Neff. 2018. Data augmentation in deep learning using generative adversarial networks. Master's thesis, Graz University of Technology.

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Hierarchical embeddings for hypernymy detection and directionality. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 233–243, Copenhagen, Denmark, September. Association for Computational Linguistics.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6341–6350. Curran Associates, Inc.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *CoRR*, abs/1904.08779.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, November.

Marek Rei, Daniela Gerz, and Ivan Vulic. 2018. Scoring lexical entailment with a supervised directional similarity network. In *ACL*.

Laura Rimell. 2014. Distributional lexical entailment by topic coherence. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 511–519, Gothenburg, Sweden, April. Association for Computational Linguistics.

Stephen Roller and Katrin Erk. 2016. Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2163–2172, Austin, Texas, November. Association for Computational Linguistics.

Stephan Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of the International Conference on Computational Linguistics*.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–363. Association for Computational Linguistics.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. 2016. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc.

Enrico Santus, Alessandro Lenci, and Qin Lu. 2014. Chasing hypernyms in vector space with entropy. In *Proceedings of the European Chapter of the Association for Computational Linguistics*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August. Association for Computational Linguistics.

A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. 2017. Learning from simulated and unsupervised images through adversarial training. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2242–2251, July.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany, August.

Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2017. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 65–75, Valencia, Spain, April. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Ran Tian, Naoaki Okazaki, and Kentaro Inui. 2017. The mechanism of additive composition. *Machine Learning*, 106(7):1083–1130.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Anthony J. Viera and Joanne M. Garrett. 2005. Understanding interobserver agreement: The kappa statistic. *Family medicine*, 37:360–3, 06.

Luke Vilnis and Andrew McCallum. 2014. Word representations via gaussian embedding. *CoRR*, abs/1412.6623.

Ivan Vulić and Nikola Mrkšić. 2018. Specialising word vectors for lexical entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1134–1145. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium, October-November. Association for Computational Linguistics.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 1015–1021, Geneva, Switzerland, Aug 23–Aug 27.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Benjamin Wilson. 2015. The unknown perils of mining wikipedia. https://blog.lateral.io/2015/06/the-unknown-perils-of-mining-wikipedia/, June.

A Supplemental Material

A.1 GANDALF Model Details

The generator and discriminator in *GANDALF* are single layer feedforward networks, with tanh activations and a dropout ratio (Srivastava et al., 2014) of 0.3. We used ADAM (Kingma and Ba, 2014) to optimise a binary cross entropy error criterion with a learning rate of 0.0002 and β values of 0.5 and 0.999. We found that *GANDALF* required quite a bit of wizardry to achieve strong performance and we found the website `https://github.com/soumith/ganhacks` very helpful. For example we applied label noise and soft labels (Salimans et al., 2016) and used a batch normalisation layer (Ioffe and Szegedy, 2015), which had the largest impact on model performance. *GANDALF* is implemented in PyTorch (Paszke et al., 2017) and we release our code on `website`.

A.2 Model Details

For our linear model we use the logistic regrssion classifier implemented in scikit-learn (Pedregosa et al., 2011). Our neural network model is 3-layer feedforward model implemented in PyTorch (Paszke et al., 2017).

We tuned the parameters of the Feedforward neural network by 10-fold cross-validation on the respective training sets, except for **LEDS**, where we chose the parameters on the basis of a model that performed well on the **Weeds** and **HP4K**. Our parameter grid consisted of activation function: {tanh, relu}, dropout: { 0.0, 0.1, 0.3 } and hidden layer sizes, where we considered { 200-200-200, 200-100-50, 200-50-30 } for Hypervec and { 600-600-600, 600-400-200, 600-300-100, 600-200-50} for word2vec. We furthermore considered 3 different aggregation functions: `diff` (Weeds et al., 2014), which simply takes the elementwise difference of the embedding pair; `asym` (Roller et al., 2014) which is the concatenation of the difference and the squared difference of the embedding pair; and `concat-asym` (Roller and Erk, 2016), which is the concatenation of the embedding pair, their difference, and their squared difference. We trained all models for 30 epochs with early stopping and used ADAM with a learning rate of 0.01 to optimise a cross entropy error criterion.

A.3 Results

Figure 5 below shows the effect of data augmentiation in terms of points of Accuracy for the logistic regression classifier per vector space model and dataset. Unlike for the higher-capacity feedforward model, data augmentation frequently causes performance to go down for the simpler linear model. This suggests that more complex models are required to fully leverage the additional information from the augmentation sets.

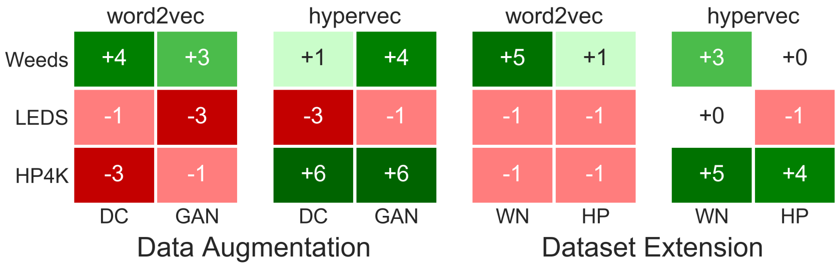

Figure 5: Effect of data augmentation and dataset extension in absolute points of accuracy on all datasets for the LR model.

Table 5 below gives an overview over the complete results for both classifier and vector space models, across all datasets. It shows the consistent positive effect of data augmentation on the more complex feedforward model in comparison to the logistic regression classifier, which is less robust to the small amounts of noise that is inevitably introduced by the automatic augmentation algorithm.

	Weeds					LEDS					HP4K					
Model	**None**	**DC**	**GAN**	**WN**	**HP**	**None**	**DC**	**GAN**	**WN**	**HP**	**None**	**DC**	**GAN**	**WN**	**HP**	**Vector Space**
LR	0.69	0.73	0.72	0.74	0.70	0.81	0.80	0.78	0.80	0.80	0.64	0.61	0.63	0.63	0.63	word2vec
FF	0.72	**0.76**	0.75	0.75	0.74	0.77	**0.83**	0.80	**0.83**	0.81	0.67	0.70	**0.71**	0.69	0.68	
LR	0.70	0.71	0.74	0.73	0.70	0.79	0.76	0.78	0.79	0.78	0.63	0.69	0.69	0.68	0.67	hypervec
FF	0.71	0.74	**0.75**	**0.75**	0.74	0.79	0.80	**0.81**	0.80	0.75	0.66	0.70	**0.72**	0.70	0.70	

Table 5: Accuracy scores for the data augmentation strategies (DC and GAN), and the two dataset extension strategies (WN and HP), and the baseline that neither uses augmentation nor extension (None). Boldfaced results denote top performance per vector space and dataset, underlined results denote improved performance in comparison to the baseline without data augmentation.