Financial Forecasting with Word Embeddings Extracted from News: A Preliminary Analysis

Luca Barbaglia, Sergio Consoli^(✉), and Susan Wang

European Commission, Joint Research Centre (JRC), Ispra, Italy
{luca.barbaglia,sergio.consoli,susan.wang}@ec.europa.eu

Abstract. News represents a rich source of information about financial agents actions and expectations. We rely on word embedding methods to summarize the daily content of news. We assess the added value of the word embeddings extracted from US news, as a case study, by using different language approaches while forecasting the US S&P500 index by means of DeepAR, an advanced neural forecasting method based on autoregressive Recurrent Neural Networks operating in a probabilistic setting. Although this is currently on-going work, the obtained preliminary results look promising, suggesting an overall validity of the employed methodology.

Keywords: DeepAR · Deep learning · Economic and financial forecasting · Neural forecasting · News analysis · Text analysis · Word embeddings

1 Introduction

Measuring the informational content of text in economic and financial news is useful for market participants to adjust their perception and expectations on the dynamics of financial markets. In this context, the incorporation in forecasting models of economic and financial information coming from news media has already demonstrated great potentials [1–3, 5]. Our endeavour is to study the predictive power of news for forecasting financial variables by leveraging on the recent advances in word embeddings [9, 21] and deep learning [17, 24] models. On the one hand, a large stand of the literature has explored the added value of word embedding technologies for forecasting applications. Shapiro et al. [25], for example, use GloVe [22] and BERT [9] word embeddings to measure economic sentiment, while Xing et al. [27] provide a review of recent works on natural language-based financial forecasting. On the other hand, recent literature has been employing neural networks for volatility forecasting [6, 18], where the volatility is a statistical measure of the dispersion of a financial asset’s returns. For example, Ramos-Pérez et al. [23] use a stacked artificial neural network to forecast volatility.

In this contribution, in particular, we show our preliminary work focusing on the prediction of the realized variance of the S&P500 index, although the

adopted methodology can be generalized to other markets and variables. To this end, we rely on word embeddings to summarize the daily content of the news contained in a data set of more than 4 million articles published in US newspapers over the period from 1st of January 2000 until 31st of December 2020. The aim is to evaluate if the combination of a richer information set including the content of economic and financial news with state-of-the-art machine learning can help in such a challenging prediction task. We assess the added value of the extracted word embeddings using different language approaches while forecasting the volatility of the S&P500 index by means of DeepAR [24], an advanced neural forecasting method based on auto-regressive Recurrent Neural Networks (RNNs) operating in a probabilistic setting. The DeepAR model is trained by adopting a rolling-window approach and employed to produce point and density forecasts by using as inputs the past time series values along with the word embeddings as additional regressors. Since our forecasting method calculates the probability attached to each forecast, the output can help investors in their decision making according to their individual risk tolerance. Our preliminary results look promising, suggesting an overall validity of the employed approach.

2 Preliminary Notions

2.1 Word Representation

For deep learning models, the text input needs to be converted to the numerical format. The simplest form is one-hot encoding [28], where each word is represented by a binary vector of size N , the size of the vocabulary, and all values are zero except for the index representing the word, marked as 1. Word embeddings improve upon one-hot-encoding by creating a lower-dimensional representation of the words such that words with similar meaning will be grouped in the vector space [16]. This is based on the idea of “distributional semantics”, where a word’s meaning is given by the words that frequently appear close by.

Word2Vec [21] and GloVe [22] are two popular algorithms for word embeddings. Word2Vec leverages the concept of a local context window where a target is surrounded by context words. It introduces the Continuous Bag-Of-Words (CBOW) algorithm [8] to predict the current word based on the surrounding context words, and the Skip-gram algorithm to predict surrounding words given the current word [16, 21]. GloVe (Global vectors for word representation) combines the concept of global matrix factorization and the local context window methods. Using the intuition that word meaning can be derived from its word co-occurrence probabilities, the model is trained to learn the weights of the word vectors by predicting global word co-occurrence counts [22].

These types of word embeddings are typically trained using a large corpus of data, and their weights are saved for future use in separate tasks. Both embedding types have 300 dimensions, and are context-free, that is, there is a one-to-one mapping between a word and its embedding representation, such that, for instance, the word “bank” has a single embedding representation in the sentence

“I am going to a bank” and “She sits by the river bank”. The embeddings for each word are, therefore, static.

2.2 Contextual Word Embeddings

Contextual word embeddings, on the other hand, take the context of each word into account when encoding words in a sentence structure. BERT, Roberta and XLM are popular contextual embedding methods based on the *transformer* architecture [4, 9], which is a recent breakthrough in the field of Natural Language Processing (NLP). The transformer was originally introduced as a means of improving neural machine translations [7, 29]. Neural machine translation methods typically consist of an encoder-decoder structure to encode a sentence into a fixed-length vector, from which a decoder generates a translation. The encoder-decoder is jointly trained to maximise the probability of a correct translation given a source sentence. Previously this was done in a sequential fashion, using sequence models, such as RNN, LSTM and GRU. The transformer instead uses a layered approach and the “attention” mechanism, to tell the model which part of the sentence to focus on while encoding the word vector. Unlike sequential models, attention can be applied to words in the sentence irrespective of the distance from the position of the word being examined, it also bypasses the need to process the sentence in a sequential manner. As such, the transformer allows sequential data such as texts in sentences to be analyzed in parallel, which not only speeds up the training process but also enables more flexibility as well as improved performance.

BERT (Bidirectional Encoder Representation from Transformer) [9] uses a multi-layer bidirectional transformer encoder architecture, and utilizes a pre-training and fine-tuning approach. Unlike Word2Vec and GloVe embeddings which are extracted and applied to separate downstream models, the most common usage of BERT model is to re-use the entire architecture in the downstream task by adding task-specific output layers and fine-tune the model with task-specific output end-to-end. BERT was the first architecture that achieved deep bi-directionality, by utilizing “Masked Language Model” (MLM) pre-training. A language model pre-training is a technique in NLP where the model is trained to predict the next word in a sentence, with the advantage being that such training does not require labelled data. In a multi-layer environment like the transformer, if a language model is trained from both left-to-right and from right-to-left, the word will inevitably “see itself” in other layers. BERT overcame this by randomly masking 15% of the input text, and train the language model to predict the masked word rather than the next word in the sentence.

DistilRoBERTa and XLM are transformer-based models that support both the fine-tuning and feature-based approaches [4]. As discussed earlier, the fine-tuning approach involves re-using the entire architecture for downstream tasks. For the feature-based approach, weights from one or more layers represent the contextual embeddings, and are extracted from the pre-trained transformer without fine-tuning any parameters. These are used as input to a subsequent deep neural network such as LSTM. Devlin et al. [9] show that the best result for the

feature-based approach is obtained by concatenating the top 4 hidden layers of BERT, achieving a result that is only slightly behind the fine-tuning approach.

2.3 Neural Forecasting

Classic techniques in economy and finance do not scale well when data are high-dimensional, noisy, and highly volatile [20]. In this complicated setting, it is not possible to rely upon standard low-dimensional strategies such as hypothesis testing for each individual variable (t -tests) or choosing among a small set of candidate models (F -test) [20]. In these cases, we are asked to provide “good” answers even if input data are extremely complex, working out of the box to recognize patterns among data and, possibly, to improve the quality of our forecasts. Following this direction, we rely on the DeepAR model [24], a neural forecasting method leveraging on previous work on deep learning and time series data [14, 17].

DeepAR’s approach is data-driven, that is, it learns a global forecasting model from historical data of all time series under consideration in the data set. The model tailors an RNN architecture into a probabilistic setting, in which predictions are not restricted to point forecasts only, but also density forecasts are produced accordingly to a user-defined distribution (e.g., negative binomial, student t , gaussian, etc.). In our case, we choose a student t -distribution in order to account for the fat-tail characteristic of the target. The outcome is more robust with respect to point forecasts alone, and uncertainty in the downstream decision-making flow is reduced by minimizing expectations of the loss function (negative log-likelihood) under the forecasting distribution. Probabilistic forecasting methods have been shown to be of crucial importance in various applications, as they -in contrast to point forecasts- enable optimal decision making under uncertainty by minimizing risk functions, that is, expectations of some loss function under the forecast distribution.

Similarly to classic RNNs, DeepAR is able to produce a mapping from input to output considering the time dimension. This mapping, however, is no longer fixed [12]. In addition to providing more accurate forecasts, DeepAR has also other advantages [24]: (i) the model infers the seasonal behavior and time series dependencies, thus reducing the tasks of manual feature engineering; (ii) the probabilistic forecasts are produced in the form of Monte Carlo samples, which are then employed to obtain consistent quantile estimates; (iii) Errors are not assumed to be Gaussian. Besides, the user chooses from a wide range of likelihood functions to better fit the properties of the data in the analysis.

3 Data

The financial time series that we aim to forecast is the annualized daily realized variance of the S&P 500 index sub-sampled from 5 min intra-day observations

obtained from the Oxford-Man Institute’s realized library¹ [11]. Following [26], we forecast the logarithmic transformation of the realized variance as it enjoys better statistical properties, while ensuring, by construction, the non-negativity of the volatility forecast. Missing data related to weekends are dropped from the target time series, giving a final number of 5,264 data points ranging from January, 3, 2000 until December, 31, 2020.

The source of economic news is obtained from a commercial provider². In our study, we consider a long time period and analyse the entire text contained in the news articles. The data set consists of more than 4 million articles, full-text, for the time period of interest for the following US outlets: The New York Times, The Wall Street Journal, The Washington Post, The Dallas Morning News, The San Francisco Chronicle, and the Chicago Sun-Times.

These newspapers are selected so as to achieve a good national as well as regional coverage. We extract sentences referring to specific economic and financial aspects, by using a keyword-based information extraction procedure with search keywords broadly related to the US economy, monetary and fiscal policies³. In order to filter out only sentences referring to US, we also use a location detection heuristic [3] assigning the location to which a sentence is referring as its most frequent named-entity location detected in the news text, and then selecting only sentences with specifically assigned location labels related to US. With this procedure, we obtain a total of over 424,578 sentences. Notice that the bank holidays might occur any day of the week, therefore the retraining step does not necessarily happen on the same day (e.g., every Friday).

4 Experiments Setup

In the first step of our experiment, we compute the word embeddings on the news data set presented in Sect. 3 relying on various embedding techniques. In particular, create a sentence embeddings by averaging individual word embeddings. We use the pre-trained Word2Vec model (“word2vec-google-news-300”) from the Python Gensim library⁴, where each word is represented by a 300 dimensional vector. The pre-processing steps include tokenisation, lower-casing, punctuation removal, stop-word removal, lemmatisation as well as the removal of out-of-vocabulary words. Then, we retrieve individual word embeddings for each word,

¹ The Oxford-Man Institute’s realized library is available at <https://realized.oxford-man.ox.ac.uk/>. The variable in the analysis corresponds to `rv5.ss`: we refer to the official website for further details on the construction of realized variances, which we employ as volatility measures.

² Dow Jones Data, News and Analytics (DNA) platform: <https://www.dowjones.com/dna/>.

³ The keywords cover various aspects of the economic activity and policy. For instance, the list includes around 300 terms, such as *inflation*, *consumer prices*, *bankruptcy*, *financial volatility*, *housing market*, *competitiveness*, *debt*, *employment*, etc.. The complete keyword list can be obtained upon request from the authors.

⁴ <https://radimrehurek.com/gensim/>.

and create sentence embeddings by taking the mean of all the word embeddings in the sentence. Similar pre-processing is applied to get the pre-trained GloVe embeddings from Gensim library. For transformer-based contextual embeddings, we use the sentence transformer library in Python⁵. All the models use mean pooling over word embeddings to obtain fixed 768 dimensional sentence embedding vectors. We consider versions of these models with and without punctuation from the text, and also considering Principal Component Analysis (PCA) over the word embeddings as a feature reduction attempt [15]⁶.

In the second step of our experiment, we use the daily average of different word embeddings as explanatory features in the DeepAR model to forecast the S&P500 log-realized variance. For our implementation, we make use of the open-source GluonTS library⁷, and experimentally adopt an architecture with 2 RNN layers having 40 LSTM cells, 500 training epochs, and a learning rate equal to 0.001⁸. We adopt a rolling window estimation technique for training and validation, with a window length equal to half of the full sample. For each window, we calculate one step-ahead forecasts. We also set a re-training step for the model equal to 5 days, meaning that every 5 consecutive data points the DeepAR model was completely retrained.

5 Preliminary Results

In this section, we show our early empirical findings on the application of DeepAR to the forecasting of the S&P 500 log-realized variance, augmented with the word embedding representation of the US news coming from the different language models presented in Sect. 2. Note that forecasting the log-realized variance of the S&P index is an extremely challenging task, as the series presents large volatility clusters. The goal is to assess whether relevant news content has some predictive power and might help in this difficult job.

Results on the comparison of the considered language models for our forecasting task using DeepAR are shown in Tables 1 and 2 for the point and density forecasts, respectively. For the evaluation, we use common time series prediction metrics, namely: mean square error (MSE), symmetric mean absolute percentage error (sMAPE), mean scaled interval score (MSIS) [19], and mean absolute scaled error (MASE). We always report the model performance relative to the one from the forecasting model without embeddings as additional regressors. Values smaller than unity indicate a better performance relative to the benchmark. On the other hand, values larger than one imply that the baseline model without word embeddings is performing better.

⁵ <https://pypi.org/project/sentence-transformers/>.

⁶ We consider the first three principal components. Future work shall address the robustness of the results of this choice.

⁷ GluonTS, available at: <https://ts.gluon.ai/>.

⁸ Future work shall address the choice of the parameters thoroughly, for instance, relying on time-series cross-validation techniques [13].

Table 1. Mean performance relative to the corresponding forecasting model with no embeddings as additional regressors: values smaller than unity (in bold) indicate a better performance relative to the benchmark.

Horizon	Embeddings	MSE	sMAPE	MSIS	MASE
$h = 1$	<i>bert_no_punctuation</i>	0.990	0.978	1.007	0.992
	<i>bert_punctuation</i>	0.983	0.983	0.992	0.989
	<i>bert_punctuation_PCA</i>	0.982	0.999	1.087	0.992
	<i>distilroberta_no_punctuation</i>	0.991	0.992	1.032	0.999
	<i>distilroberta_punctuation</i>	0.993	0.989	1.027	0.997
	<i>glove_no_punctuation</i>	1.005	0.998	1.025	0.995
	<i>xlm_punctuation</i>	1.077	1.023	1.082	1.030

Table 1 reports the forecasting performance across all windows for the point forecasts. From the table, we denote that there is not yet a clear superiority of a word embedding approach with respect to the others. They perform comparably well, providing an added value relative to the corresponding forecasting model without embeddings. There is an exception with *xlm_punctuation*, which attains worse performances relative to the corresponding forecasting model without embeddings regardless of the metric; probably XLM training parameters should be better fine-tuned in future experiments. We can also note that there is not a clear distinction between the word embedding models with and without punctuation, although a slight superiority is obtained when punctuation is considered⁹.

From this early experiment, we also see that the feature reduction attempt in BERT using PCA does not provide benefits. We plan to try alternative approaches, like, e.g., employing hierarchical clustering and selecting only embedding features closer to cluster centroids. We believe that feature reduction can provide performance improvements, even though at the moment we are not getting any clear experimental proof. We test the significance of the forecast gains relying on the conditional predictive ability test by Giacomini and White [10], which finds that only the *bert_punctuation* performs significantly better than the benchmark when considering the sMAPE metric at the 90% confidence level.

Table 2 reports the quantile losses at the 0.1, 0.5 and 0.9 quantiles. The best performance for the highest quantiles is obtained by the BERT models with and without punctuation and by XLM, while the rest of the models produce a worse performance with respect to the model without embeddings. This result is something we can expect, given that obtaining results for rare events provided by high quantiles is quite hard and unpredictable. However, BERT models are able to obtain acceptable performance also in these cases, confirming a good generalization capability of the underlying model. As it regards the median forecast,

⁹ For some models in the table we report indeed only results with punctuation, given their results without punctuation are the same.

Table 2. Quantile losses for $\tau = 0.1, 0.5$ and 0.9 quantiles, relative to the corresponding forecasting model with no embeddings as additional regressors: values smaller than unity (in bold) indicate a better performance than the benchmark.

Horizon	Embeddings	$\tau = 0.1$	$\tau = 0.5$	$\tau = 0.9$
$h = 1$	<i>bert_no_punctuation</i>	1.023	0.992	0.988
	<i>bert_punctuation</i>	1.029	0.989	0.966
	<i>bert_punctuation_PCA</i>	1.064	0.993	1.032
	<i>distilroberta_no_punctuation</i>	1.029	0.999	1.012
	<i>distilroberta_punctuation</i>	1.022	0.997	1.011
	<i>glove_no_punctuation</i>	0.990	0.995	1.031
	<i>xlm_punctuation</i>	1.067	0.991	0.978

all models perform better than the benchmark, while only GloVe attains a forecast gain for the lowest quantile. The Giacomini and White [10] test indicates that only *bert_punctuation* attains a significantly better performance than the benchmark when considering a 95% confidence level. The poor performance in the 0.1 quantile might be explained by the logarithmic transformation of the target variable: in future research, we plan to experiment further on this issue.

Word embeddings extracted from economic news generally provide improvements when the DeepAR model is combined with them. This suggests that the content extracted from news contain some predictive power for the target to forecast. When these features are added to the corresponding DeepAR model, the results improve in terms of the considered metrics.

6 Conclusion and Overlook

Word embeddings extracted from news appear to have predictive power for the forecasting exercise of the S&P 500 log-realized variance. DeepAR manages to achieve good prediction results, performing better when the news embeddings are included in the model. We believe that the combination of these cutting-edge technologies has a high potential for economic and financial forecasting applications. The obtained results, although preliminary, look encouraging.

In the future steps of this project, we plan to attempt increasing the forecasting performance of our approach by fine-tuning the pre-trained language models directly with the considered target. In addition, we plan to use other cutting-edge forecasting methods from machine learning in order to have a comparison with respect to the results obtained by the DeepAR model. Future computational experiments will include statistical testing of the significance of the forecast gains. Furthermore, we might also consider novel sentence embedding methods, where a sentence transformer is included and adds a pooling operation to the output of the transformers of the contextual word embedding methods (BERT, RoBERTa or XLM) to derive fixed-size sentence embeddings. The weights of the transformers are shared, so the resulting sentence embeddings are semantically meaningful and can be compared using cosine similarity. Finally, further work

might explore the forecasting performance of the proposed methodology when considering the various underlying assets that are included in the S&P500.

References

1. Apergis, N., Lau, M.C.K., Yarovaya, L.: Media sentiment and CDS spread spillovers: evidence from the GIIPS countries. *Int. Rev. Financ. Anal.* **47**(C), 50–59 (2016)
2. Barbaglia, L., Consoli, S., Manzan, S.: Exploring the predictive power of news and neural machine learning models for economic forecasting. In: Bitetta, V., Bordino, I., Ferretti, A., Gullo, F., Ponti, G., Severini, L. (eds.) *MIDAS 2020. LNCS (LNAI)*, vol. 12591, pp. 135–149. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-66981-2_11
3. Barbaglia, L., Consoli, S., Manzan, S.: Forecasting with economic news. SSRN Working paper 3698121 (2021)
4. Barua, A., Thara, S., Premjith, B., Soman, K.P.: Analysis of contextual and non-contextual word embedding models for Hindi NER with web application for data collection. In: Garg, D., Wong, K., Sarangapani, J., Gupta, S.K. (eds.) *IACC 2020. CCIS*, vol. 1367, pp. 183–202. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-0401-0_14
5. Beetsma, R., Giuliodori, M., de Jong, F., Widijanto, D.: Spread the news: the impact of news on the European sovereign bond markets during the crisis. *J. Int. Money Financ.* **34**, 83–101 (2013)
6. Bucci, A.: Realized volatility forecasting with neural networks. *J. Financ. Economet.* **18**(3), 502–531 (2020)
7. Clinchant, S., Jung, K.W., Nikoulina, V.: On the use of BERT for neural machine translation. In: *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pp. 108–117. Association for Computational Linguistics (2019)
8. Csurka, G., Dance, C.R., Willamowski, L.J., Bray, C.: Visual categorization with bags of keypoints. In: *Workshop on Statistical Learning in Computer Vision, ECCV*, pp. 1–22 (2004)
9. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of NAACL-HLT 2019 - The 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 4171–4186 (2019)
10. Giacomini, R., White, H.: Tests of conditional predictive ability. *Econometrica* **74**(6), 1545–1578 (2006)
11. Heber, G., Lunde, A., Shephard, N., Sheppard, K.: Oxford-man institute’s realized library. Oxford-Man Institute, University of Oxford, UK (2009). <https://realized.oxford-man.ox.ac.uk/>
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997)
13. Hyndman, R.J., Athanasopoulos, G.: *Forecasting: Principles and Practice*, 3rd edn. OTexts, Melbourne (2021)
14. Januschowski, T., et al.: Criteria for classifying forecasting methods. *Int. J. Forecast.* **36**(1), 167–177 (2020)
15. Jolliffe, I.T., Cadima, J.: Principal component analysis: a review and recent developments. *Philos. Trans. Roy. Soc. A Math. Phys. Eng. Sci.* **374**(2065) (2016)

16. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: 32nd International Conference on Machine Learning, ICML 2015, vol. 2, pp. 957–966 (2015)
17. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
18. Liu, Y.: Novel volatility forecasting using deep learning-long short term memory recurrent neural networks. *Expert Syst. Appl.* **132**, 99–109 (2019)
19. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Predicting/hypothesizing the findings of the M4 competition. *Int. J. Forecast.* **36**(1), 29–36 (2020)
20. Marwala, T.: *Economic Modeling Using Artificial Intelligence Methods*. Springer, London (2013). <https://doi.org/10.1007/978-1-4471-5010-7>
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013 (2013)
22. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: *Proceedings of EMNLP 2014 - Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543 (2014)
23. Ramos-Pérez, E., Alonso-González, P.J., Núñez-Velázquez, J.J.: Forecasting volatility with a stacked model based on a hybridized artificial neural network. *Expert Syst. Appl.* **129**, 1–9 (2019)
24. Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T.: DeepAR: probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.* **36**(3), 1181–1191 (2020)
25. Shapiro, A.H., Sudhof, M., Wilson, D.J.: Measuring news sentiment. *J. Econ.* (2020)
26. Wilms, I., Rombouts, J., Croux, C.: Multivariate volatility forecasts for stock market indices. *Int. J. Forecast.* **37**(2), 484–499 (2021)
27. Xing, F.Z., Cambria, E., Welsch, R.E.: Natural language based financial forecasting: a survey. *Artif. Intell. Rev.* **50**(1), 49–73 (2017). <https://doi.org/10.1007/s10462-017-9588-9>
28. Zhang, Y., Jin, R., Zhou, Z.-H.: Understanding bag-of-words model: a statistical framework. *Int. J. Mach. Learn. Cybern.* **1**(1–4), 43–52 (2010)
29. Zhu, J., et al.: Incorporating BERT into neural machine translation. In: *International Conference on Learning Representations (ICLR)* (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.