Hyperparameter Optimization of Semi-Supervised Sentiment Annotation Model on Marketplace Dataset

Nur Heri Cahyana[1*], Yuli Fauziah[2], Wisnalmawati[3], Agus Sasmito Aribowo[4]
*Corresponding author email: nur.hericahyana@upnyk.ac.id
[1,2]Department of Informatics, National Development University of Veteran Yogyakarta, Yogyakarta, Indonesia
[3]Faculty of Economics and Business, National Development University of Veteran Yogyakarta, Yogyakarta, Indonesia
[4]Student at Faculty of Information & Communication Technology, University Teknikal Malaysia, Melaka, Malaysia

Abstract. Hyperparameter optimization in semi-supervised learning (SSL) models for sentiment analysis is the process of adjusting the machine learning model's parameters to enhance the performance of the SSL system. Hyperparameter tuning aims to maximize the accuracy, precision, recall, and F1-Score of each machine learning component in the SSL model. The SSL model subjected to hyperparameter tuning is an outcome of previous research specifically designed for the automatic annotation process in sentiment analysis. This paper aims to enhance the performance of machine learning within the SSL model, which includes the Support Vector Machine (SVM) algorithm and Random Forest classifier. The parameter in the Random Forest classifier is the number of trees (estimators). For the SVM method, the parameter in question is the value of 'C.' The hyperparameter testing employed the Random Search method. The dataset used consists of customer comments from a marketplace. The final result is hyperparameter tuning has succeeded in improving the sentiment annotation processing capabilities of SSL on the marketplace dataset. Improving the performance of sentiment annotation can be done by adjusting the C parameter in the SVM method and the number of estimator parameters in the Random Forest (RF) Classifier.

Keywords: hyperparameter tuning, machine learning, sentiment analysis.

1 Introduction

Hyperparameter optimization in semi-supervised sentiment analysis models is the process of adjusting parameters that cannot be learned by the model independently (referred to as hyperparameters) to enhance the model's performance in identifying or analyzing sentiment with only a small portion of labeled data. This is a critical stage in the development of effective machine-learning models. Hyperparameter tuning aims to achieve better performance in hate speech analysis tasks by maximizing accuracy, precision, recall, or other relevant evaluation metrics. The model was subjected to hyperparameter tuning results from previous research specifically designed for the automatic annotation process in sentiment analysis using semi-supervised learning.

Sentiment annotations typically involve an expert who categorizes documents into two polarities: positive and negative[1]–[9]. Automated annotation utilizes minimal training data to

B. Sobirov et al. (eds.), *Proceedings of the 2nd International Conference on Advanced Research in Social and Economic Science (ICARSE 2023)*, Advances in Social Science, Education and Humanities Research 842,
https://doi.org/10.2991/978-2-38476-247-7_18

construct the model, generating automatic annotations for all datasets used in the study. Semi-supervised text annotations typically utilize a sentiment lexicon to manually label and refine a sample of annotated data by the annotator. Consequently, this process is time- consuming[10]. Semi-supervised learning has been developed for automatic annotation [10], sentiment, and emotion analysis [11]. One well-known SSL model is the AraSenCorpus framework. AraSenCorpus can annotate Arabic text corpus using FastText and Long-Short Term Memory (LSTM) neural network learning approaches based on unlabeled Twitter data, reducing the need for manual annotation efforts. AraSenCorpus can identify sentiment analysis with two polarities (negative and positive) or three polarities (positive, neutral, and negative)[10]. his framework can improve sentiment classification accuracy by more than 7% compared to the manual process. However, the framework does not specify the type of classifier and vectorizer used and does not incorporate hyperparameter tuning.

Several studies also apply semi-supervised text annotation in English documents, as seen in[11]. Annotations are done using various machine learning approaches, combining supervised and unsupervised methods. Three experts perform initial data annotation. The annotation process, sentiment analysis, and emotion analysis involve several machine learning techniques such as SVM, Random Forest (RF), and Naïve Bayes (NB)[11]. While this method can analyze sentiment and emotion concurrently, the average accuracy of emotion classification is notably low. However, this research does not include hyperparameter tuning. Other studies have used Twitter datasets, including LSTM, SVM, and convolutional neural networks (CNN), to annotate the Saudi dialect [18]. However, this research does not incorporate hyperparameter tuning, resulting in low classifier performance for SVM across all thresholds. Therefore, we propose enhancements by employing hyperparameter tuning in semi-supervised learning for sentiment annotation using the SVM and Random Forest methods combined with TF-IDF to address the low-performance issues in the annotation process.

This research will further discuss several points: the method of annotation and sentiment analysis using semi-supervised learning described in Section 2. Next, Section 3 presents the results and discussion of the automatic annotation process and sentiment detection. Finally, we summarize and conclude the research conducted in Section 4

2 Research Methods

2.1 Hyperparameter tuning strategy

This study implements hyperparameter tuning in semi-supervised learning to conduct the sentiment annotation process on a marketplace dataset. The annotation process is performed to interpret words by labeling the data. Two machine learning models, Random Forest and SVM, will be subjected to hyperparameter tuning. In the case of SVM, this study tunes the C value. In the case of Random Forest, this study adjusts the number of decision trees (estimators). The hyperparameter tuning process utilizes the random search method. The random search method was selected due to its efficiency and favorable results. Therefore, this study implements the RandomizedSearchCV method from Scikit-Learn. The annotations in this study involve over 25,000 data, with a 20:80 split between training and test data, using a threshold value of 0.5. This annotation process will process text from the dataset by scraping marketplace comments. The steps involved in annotating sentiment are illustrated in Figure 1.

The initial step entails ingesting a dataset divided into two categories: labeled data and unlabeled data. Labeled data comprises information annotated by experts and is employed for training in this annotation process. The labeled data is further divided into Data Training and Data Testing subsets. Data Training, Data Testing, and the Unlabeled data are preprocessed to standardize the text for feature extraction using the Term Frequency-Inverse Document Frequency (TF-IDF) method.

The results of TF-IDF feature extraction are utilized for training the SVM and Random Forest methods, constituting a meta-learning model for the automatic annotation process. The SVM and Random Forest models undergo hyperparameter tuning before annotating unlabeled documents. The baseline condition refers to a scenario in which the models, following hyperparameter tuning, are used to evaluate labeled training data by experts. The final state is one in which the models, having undergone hyperparameter tuning, are employed to assess labeled training data by experts and data that the machine has already annotated.

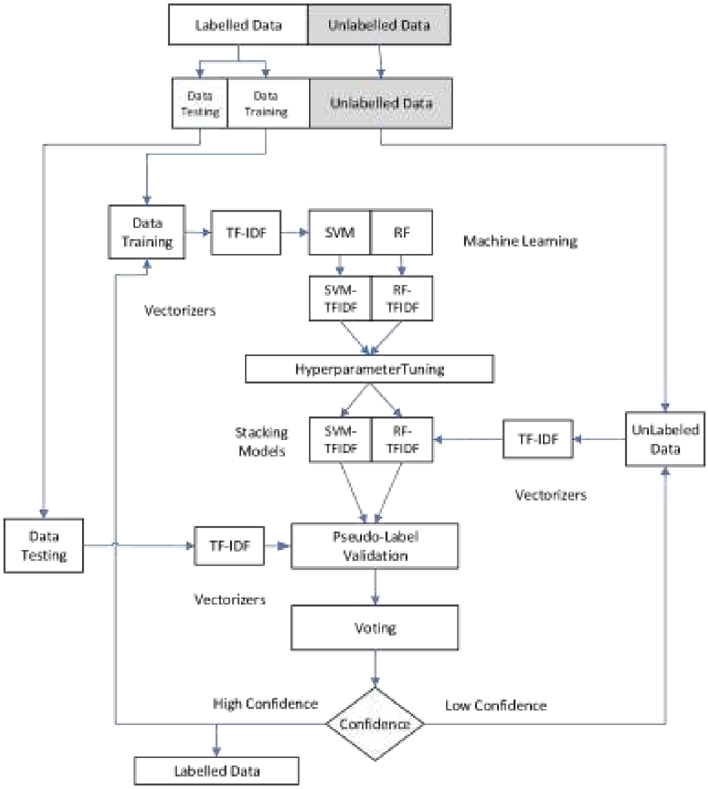

Fig. 1. Hyperparameter Tuning on semi-supervised sentiment annotation (SSL) model.

The automatic annotation results become a follow-up process to check datasets that have not been annotated. This process predicts the label based on a combination of vectorization and machine learning, with the expected value determining its accuracy and F1-Score. The labeling process determines the label using a voting process by calculating the weight value from machine learning performance, both for the positive and negative polarity. If the polarity score has a value that exceeds the threshold (0.5), then the dataset and its annotations are sent to the training data. However, if not, the data will go through the following process. The process cycle lasts ten rounds. If it exceeds the process, the remaining unlabeled dataset is manually annotated and combined with the training data.

2.2 Marketplace dataset

In this study, we used marketplace opinions datasets from social media containing 25000 opinions about market transactions. Marketplace is a dataset for binary sentiment classification with positive and negative labels. Marketplace datasets need to be analyzed consisting of words, numbers, and special symbols. Some processes for structuring the data go through several stages, such as tokenizing, converting to a small case, removing a number, removing stop words, removing all non-alphabetic characters and punctuation, and stemming.

Text in the dataset is unstructured data, so it is recommended to do preprocessing [12] to get optimal and more structured results [13]. This preprocessing can reduce unnecessarily duplicated, irrelevant, and noisy data [14]. This stage is carried out before the feature extraction and classification process.

2.3 Feature extraction (TF-IDF)

The result of preprocessing is the standard text used in feature extraction. This feature extraction aims to convert the input data into a meaningful feature [15]. This research uses TF-IDF and Word2Vec feature extraction. TF-IDF was carried out to perform word weighting, which gave different weights [16] for each term in the document based on the frequency per document and all documents [17]. TF-IDF perform well which is applied in this study [13].

2.4 SVM classifier

The results of the feature extraction of each method will be classified using the SVM Classifier method. This method applies linear classifiers and non-linear cases to find the best hyperplane in the classification [18], [19]. This method also refers to statistical learning methods with accuracy values depending on specific functions and parameters[20]. The detection process in this study uses the calculated TF-IDF for each feature parameter. These parameters are input to the SVM classifier's training and testing process.

SVM is often used in text classification methods because of its superior performance [21] and sentiment detection [22]. SVM is the most popular classification method, with superior performance in inserting the dimensions of the input feature space [23]. This method aims to find the most suitable classification function to distinguish between members of the two classes in the training data [24]. In addition, this method will separate data points and evaluate each category of data points into a hyperplane, as well as maximize the margin between support vectors because separating all classes is required [22].

2.5 Random forest

Random Forest (RF) is the development of a Decision Tree. Random Forest builds many trees in the same steps as a Decision Tree. Random Forest reduces the risk of overfitting by introducing randomness by building multiple trees, bootstrapping, and splitting nodes using the best split among a random subset of features selected at every node. An example of the process of making many trees is in Figure 2.9. Each tree provides the results of classification. The final classification is the most classes produced from these trees (majority class) [25].

3 Results and Discussion

In this section, we present the results and discussion into 4 parts: dataset crawling, simulation of the annotation process using Python, and classification results. Each section describes the process, results, and discussions based on the experiments that have been carried out.

3.1 Crawling datasets and scenarios

The dataset was obtained by crawling commentary data from a marketplace, which covered various topics such as marketplace service, product quality, customer satisfaction, and after-sales service. A total of 25,000 comments were collected for use in the study. The crawling process was performed directly by extracting customer comments from the marketplace using Python programming. Out of the obtained data, 20% (5,000 data points) were subjected to annotation. The annotation process involved classifying comments into two categories: positive and negative. The initial manually annotated data was employed for both training and testing, following a 20:80 split. The modeling process used a threshold value of 0.5 to determine the assigned annotations.

3.2 SSL hyperparameter simulation results using python

The annotation process commenced with a review of the 5,000 datasets that had been expertly labeled as either positive or negative. The annotated dataset, created by expert annotators, was partitioned into two subsets: training data and testing data, in an 80:20 ratio. This partitioning utilized the stratified sampling technique to maintain proportional representation of each data class within the ratio. Consequently, 4,000 records were designated for the training data, while 1,000 records were allocated to the testing data. Subsequently, the annotation process continued by processing an additional 20,000 unlabeled datasets.

3.2.1 Iteration 1

3.2.1.1 Hyperparameter in Iteration

The subsequent step involves configuring the hyperparameters for each machine learning component used within the model. The hyperparameters for SVM and Random Forest are determined using the training data that has undergone the TF-IDF vectorization processes. The hyperparameter tuning process employs the Random Search method. The outcomes of the hyperparameter tuning process for all initial-stage machine learning components are presented in Table 1.

Table 1. SSL model hyperparameter result for baseline and iteration 1

	Without Hyperparameter Tuning		With Hyperparameter Tuning		
Machine Learning	**Accuracy**	**F1-Score**	**Best Parameter**	**Accuracy**	**F1-Score**
Random Forest	0.795	0.803	Estimators: 100	0.803	0.823
SVM	0.818	0.822	C: 4.28	0.837	0.847

3.2.1.2 Baseline condition

The subsequent step involves calculating the performance of the model under baseline conditions. Performance is measured by annotating the testing data and subsequently compari ng the actual labels with the predicted labels. The baseline process demonstrates that the mode l annotates the testing data. These performance metrics will serve as a reference for evaluating the performance of the SSL-Model in annotating unannotated datasets. Based on Table 1 abov e, it can be seen that at the baseline stage, SSL performance has increased in all measurement metrics, both Accuracy and F1-Score after going through the hyperparameter tuning process. I mprovements have also been experienced in all types of machine learning.

3.2.1.3 Annotation in Iteration 1

The next step involves the annotation process of unlabelled data using the SSL-Model. The conditions of the training data and unlabelled dataset prior to the annotation process are as follows :

```
The Number of Annotated Dataset   (record) :
4000  The  Number  of  Unannotated  Dataset
(record) : 20000 The Number of Total Dataset
(record) : 24000
```

The conditions of the dataset after the annotation process are as follows:

```
The Number of Annotated Dataset
                                        (record
) : 19385 The Number of Unannotated Dataset
                                        (record
) : 4615 The Number of Total Dataset
                                        (record
) : 24000
```

From a total of 20000 unlabeled data, successful labeling has been achieved for a count of 15385 data. The annotation process leaves 4615 unlabeled data remaining. The total number of labeled data (by experts and the machine) eventually reaches 19385 data. Hence, the annotation process continues into Iteration 2, utilizing the training data consisting of the 19385 documents.

3.2.2 IIteration 2

3.2.2.1 Hyperparameter in Iteration 2

The subsequent step involves configuring the hyperparameters for each machine learning component used within the SSL-Model for the second iteration. The hyperparameters for all machine learning components are determined using the updated training data, which has been reprocessed using TF-IDF vectorization. The hyperparameter tuning process continues to employ the Random Search method. The outcomes of the hyperparameter tuning process for the second iteration are provided in Table 2.

Table 2. SSL-model hyperparameter result in 2nd iteration.

	Without Hyperparameter Tuning		With Hyperparameter Tuning		
Machine Learning	**Accuracy**	**F1-Score**	**Best Parameter**	**Accuracy**	**F1-Score**
Random Forest	0.835	0.857	Estimators: 110	0.859	0.862
SVM	0.885	0.924	C: 78.475	0.938	0.942

Based on Table 2 above, it can be seen that at the baseline stage, SSL performance has increased in all measurement metrics, both Accuracy and F1-Score after going through the hy perparameter tuning process. Improvements have also been experienced in all types of machin e learning.

3.2.2.2 Annotation in Iteration 2

The subsequent step involves the annotation process of remaining unlabelled data from the previous iteration using the SSL-Model. The conditions of the data before the annotation process in the second iteration are as follows:

```
The Number of Annotated Dataset     (record) :
19385  The  Number  of  UnAnnotated  Dataset
(record) :   4615 The Number of Total Dataset
                                      (record
) : 24000
```

The conditions of the dataset after the annotation process are as follows:

```
The Number of Annotated Dataset     (record) :
24000  The  Number  of  UnAnnotated  Dataset
(record) :                                     0

The Number of Total Dataset          (record) : 24000
```

Based on the results above, all unlabelled datasets have been successfully labeled. Therefore, the annotation process is concluded, and the validation of the annotation outcomes will be pursued.

3.2.2.2 Validation of annotation result

The next step involves validating the annotation outcomes. The validation process is carried out by utilizing all labeled training data (24000 records) to create the SSL-Model and subsequently applying it to label the testing data. When compared to the baseline, the results are as presented in Table 3.

Table 3. Comparison of baseline performance with final performance

Performance	Without Hyperparameter Tuning		With Hyperparameter Tuning	
	Baseline	Final	Baseline	Final
F1-Score	0.823	0.819	0.828	0.842
Precision	0.821	0.813	0.838	0.842
Recall	0.814	0.835	0.826	0.845
Accuracy	0.813	0.807	0.824	0.825

Table 3 indicates a slight performance increase between the baseline and final conditions. This is likely due to the quality of labeling by the SSL-Model formed using training data and pseudo-labels, which is not as accurate as the labeling by the SSL-Model trained with data that had been expert-labeled. This research acknowledges that such a phenomenon is reasonable if the performance drop is not substantial, ensuring that the labeling quality remains satisfactory. The annotation outcomes by the SSL-Model result in a dataset with labels indicating fanatic and non-fanatic, as shown in Figure 2.

Labell
ed as Positive

Unl
abelled
Docum
ent

Labelled
as Negative

Fig. 1. Result of fanaticism annotation by SSL model

4 Conclusions

Hyperparameter tuning has succeeded in improving the sentiment annotation processing capabilities of SSL on the marketplace dataset. The performance of sentiment annotation can be improved by adjusting the C parameter in the SVM method and the number of estimator parameters in the Random Forest (RF) Classifier. Hyperparameter tuning in SVM and RF is recommended because it provides more accurate results in the validation mode process. The best accuracy of the two machine learning is found in the SVM model. In the future, it is necessary to improve the optimization of the SSL annotation and validation process using other machine learning methods, such as CNN or LSTM.

Acknowledgments

Special Thank for LPPM Universitas Pembangunan Nasional Veteran Yogyakarta, was supported this publication.

References

[1] S. Aman and S. Szpakowicz, "Identifying expressions of emotion in text," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4629 LNAI, no. September, pp. 196–205, 2007, doi: 10.1007/978-3-540-74628-7_27.

[2] A. Krouska, C. Troussas, and M. Virvou, "The effect of preprocessing techniques on Twitter sentiment analysis," in Proceedings of the 7th International Conference on Information, Intelligence, Systems and Applications (IISA), 2016, doi: 10.1109/IISA.2016.7785373.

[3] J. Savigny and A. Purwarianti, "Emotion classification on Youtube comments using word embedding," in International Conference on Advanced Informatics: Concepts, Theory and Applications, 2017, pp. 1–5, doi: 10.1109/ICAICTA.2017.8090986.

[4] A. M. Ningtyas and G. B. Herwanto, "The Influence of Negation Handling on Sentiment Analysis in Bahasa Indonesia," in Proceedings of the 5th International Conference on Data and Software Engineering (ICoDSE), 2018, pp. 1–6, doi: 10.1109/ICODSE.2018.8705802.

[5] K. Mulcrone, "Detecting Emotion in Text," p. 6, 2012.

[6] W. C. F. Mariel, S. Mariyah, and S. Pramana, "Sentiment analysis: A comparison of deep learning neural network algorithm with SVM and naïve Bayes for Indonesian text," in Journal of Physics: Conference Series, vol. 971, no. 1, 2018, doi: 10.1088/1742-6596/971/1/012049.

[7] T. Sutabri, A. Suryatno, D. Setiadi, and E. S. Negara, "Improving Naïve Bayes in Sentiment Analysis For Hotel Industry in Indonesia," in Proceedings of the 3rd International Conference on Informatics and Computing (ICIC), 2018, pp. 1–6, doi: 10.1109/IAC.2018.8780444.

[8] M. Lailiyah, S. Sumpeno, and I. K. E. Purnama, "Sentiment analysis of public complaints using lexical resources between Indonesian sentiment lexicon and sentiwordnet," in Proceedings of the 2017 International Seminar on Intelligent Technology and Its Application (ISITIA), 2017, vol. 2017-Janua, pp. 307–312, doi: 10.1109/ISITIA.2017.8124100.

[9] U. Makhmudah, S. Bukhori, J. A. Putra, and B. A. B. Yudha, "Sentiment Analysis of Indonesian Homosexual Tweets Using Support Vector Machine Method," in Proceedings of the 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), 2019, pp. 183–186, doi: 10.1109/ICOMITEE.2019.8920940.

[10] A. Al-Laith, M. Shahbaz, H. F. Alaskar, and A. Rehmat, "Arasencorpus: A semi-supervised approach for sentiment annotation of a large arabic text corpus," in Applied Sciences (Switzerland), vol. 11, no. 5, 2021, doi: 10.3390/app11052434.

[11] V. Balakrishnan, P. Y. Lok, and H. Abdul Rahim, "A semi-supervised approach in detecting sentiment and emotion based on digital payment reviews," in Journal of Supercomputing, vol. 77, no. 4, pp. 3795–3810, 2021, doi: 10.1007/s11227-020-03412-w.

[12] D. A. Abduljabbar and N. Omar, "Exam questions classification based on Bloom's taxonomy cognitive level using classifiers combination," in Journal of Theoretical and Applied Information Technology, vol. 78, no. 3, p. 447, 2015.

[13] D. E. Cahyani and I. Patasik, "Performance comparison of TF-IDF and Word2Vec models for emotion text classification," in Bulletin of Electrical Engineering and Informatics, vol. 10, no. 5, pp. 2780–2788, Oct. 2021, doi: 10.11591/eei.v10i5.3157.
[14] M. Mohammed and N. Omar, "Question classification based on Bloom's taxonomy cognitive domain using modified TF-IDF and word2vec," in PLOS ONE, vol. 15, no. 3, p. e0230442, Mar. 2020, doi: 10.1371/journal.pone.0230442.
[15] H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text feature extraction based on deep learning: a review," in EURASIP Journal on Wireless Communications and Networking, vol. 2017, no. 1, p. 211, Dec. 2017, doi: 10.1186/s13638-017-0993-1.
[16] S. Saifullah, Y. Fauziyah, and A. S. Aribowo, "Comparison of machine learning for sentiment analysis in detecting anxiety based on social media data," in Jurnal Informatika, vol. 15, no. 1, p. 45, 2021, doi: 10.26555/jifo.v15i1.a20111.
[17] Y. Fauziah, S. Saifullah, and A. S. Aribowo, "Design Text Mining for Anxiety Detection using Machine Learning based-on Social Media Data during COVID-19 pandemic," in Proceeding of LPPM UPN "Veteran" Yogyakarta Conference Series 2020–Engineering and Science Series, vol. 1, no. 1, pp. 253–261, 2020, doi: 10.31098/ess.v1i1.117.
[18] S. Saifullah and A. P. Suryotomo, "Identification of chicken egg fertility using SVM classifier based on first-order statistical feature extraction," in ILKOM Jurnal Ilmiah, vol. 13, no. 3, 2021.
[19] S. Saifullah and R. Drezewski, "Non-Destructive Egg Fertility Detection in Incubation Using SVM Classifier Based on GLCM Parameters," in Procedia Computer Science, vol. 207C, pp. 3248–3257, 2022.
[20] S. J. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach." New Jersey: Pearson Education, Inc., 2010.
[21] Y. Li, Y. Liu, Z. Zhu, and P. Liu, "Exploring semantic awareness via graph representation for text classification," in Applied Intelligence, May 2022, doi: 10.1007/s10489-022-03526-z.
[22] D. A. Pisner and D. M. Schnyer, "Support vector machine," in Machine Learning, pp. 101–121, 2020, doi: 10.1016/B978-0-12-815739-8.00006-7.
[23] Y. Hu, J. Ding, Z. Dou, and H. Chang, "Short-Text Classification Detector: A Bert-Based Mental Approach," in Computational Intelligence and Neuroscience, vol. 2022, pp. 1–11, Mar. 2022, doi: 10.1155/2022/8660828.
[24] B. AlBadani, R. Shi, and J. Dong, "A Novel Machine Learning Approach for Sentiment Analysis on Twitter Incorporating the Universal Language Model Fine-Tuning and SVM," in Applied System Innovation, vol. 5, no. 1, p. 13, Jan. 2022, doi: 10.3390/asi5010013.
[25] L. Breiman, "Random Forests," vol. 45. Netherlands: Kluwer Academic Publishers, 2001.