Identifikasi Bahasa Otomatis dengan *Machine Learning*

Galuh Tunggadewi Sahid

Universitas Indonesia, Depok, Jawa Barat, Indonesia 16424

Abstract

Identifikasi bahasa merupakan salah satu permasalahan penting dalam *natural language processing*. Mengidentifikasi bahasa dari sebuah dokumen dapat dilakukan dengan memanfaatkan *machine learning.* Pada eksperimen ini, fitur-fitur yang terdapat dari sebuah teks akan diekstrak berdasarkan n-gram dari kata. Selanjutnya, dokumen akan diklasifikasikan berdasarkan fitur-fitur yang sudah diekstrak sebelumnya dengan memanfaatkan berbagai metode klasifikasi yaitu *decision tree, neural network,* dan *Bayesian network*. Setelah mengkombinasikan berbagai model n-gram dan metode klasifikasi, didapatkan hasil bahwa klasifikasi dengan menggunakan *Bayesian network* dengan jenis ekstraksi fitur boolean 1-gram memberikan hasil yang terbaik dalam mengidentifikasi bahasa secara otomatis.

Keywords: identifikasi bahasa, *machine learning,* klasifikasi

1. Pendahuluan

Identifikasi bahasa secara otomatis merupakan salah satu permasalahan *natural language processing*. Dengan kemampuan identifikasi bahasa secara otomatis, maka banyak permasalahan lain yang dapat diselesaikan, salah satunya adalah melakukan personalisasi terhadap hasil pencarian berdasarkan bahasa dari kata kunci pencarian (Stiller, 2010). Hal ini penting mengingat peningkatan penggunaan Internet saat ini diiringi pula dengan peningkatan dari jumlah informasi yang tersedia, di mana seluruh informasi ini memiliki konten yang beragam dan juga terdapat pada bahasa yang berbeda-beda. Peningkatan jumlah informasi menimbulkan tantangan agar suatu informasi dapat ditemukan dengan mudah di tengah banyaknya informasi yang tersedia, sehingga kemampuan untuk mengidentifikasi bahasa secara otomatis sangat dibutuhkan.

Di sisi lain, perkembangan *machine learning* yang semakin pesat membuat *machine learning* dapat digunakan dalam berbagai aplikasi, salah satunya adalah klasifikasi. Klasifikasi dengan menggunakan *machine learning* seringkali memberikan hasil yang lebih akurat dibandingkan dengan klasifikasi menggunakan peraturan-peraturan yang dibuat oleh manusia karena sifat *data-driven* yang dimiliki oleh *machine learning* (Schapire, n.d.).

Didorong oleh hal tersebut, penulis melakukan eksperimen untuk mengidentifikasi bahasa dengan klasifikasi menggunakan *machine learning*. Eksperimen dilakukan pada berbagai kombinasi metode ekstraksi fitur serta metode klasifikasi yang berbeda-beda. Kemudian, dilakukan perbandingan hasil klasifikasi antara satu kombinasi dengan kombinasi yang lainnya berdasarkan jumlah sampel yang dapat diidentifikasi dengan bahasa yang benar.

Dataset untuk eksperimen diberikan dalam bentuk sebuah berkas XML yang berisi dokumen-dokumen berbentuk teks yang telah diberikan label bahasa masing-masing. Untuk eksperimen 1, 2, 3, dan 4, penulis akan menggunakan *dataset* yang telah diberikan sebagai soal. Pada eksperimen kelima, penulis akan menggunakan *dataset* yang penulis kumpulkan sendiri yang prosesnya akan dijelaskan lebih lanjut.

Untuk dapat memulai proses klasifikasi, perlu dilakukan ekstraksi fitur dalam bentuk n-gram dari kata-kata yang terdapat pada dokumen. Ekstraksi fitur dalam bentuk n-gram dilakukan baik dengan perhitungan boolean maupun frekuensi. Fitur-fitur yang dihasilkan akan diklasifikasikan dengan menggunakan algoritma *machine learning* yaitu *decision tree, neural network, dan Bayesian network* dengan menggunakan perangkat lunak Weka.

Pada bagian 2, laporan ini akan membahas mengenai *classifier* yang digunakan, teknik *cross validation* yang digunakan, serta metode untuk mengevaluasi *classifier*. Bagian 3 akan membahas representasi masalah dalam *machine learning* serta proses pengolahan *dataset* sebelum diproses oleh perangkat lunak Weka. Bagian 4 akan memaparkan hasil dari eksperimen yang telah dilakukan beserta analisisnya.

2. Latar Belakang

2.1. Klasifikasi

Pada *machine learning,* klasifikasi merupakan permasalahan dalam mengidentifikasi kategori dari sebuah observasi baru berdasarkan sebuah *training set* berisi observasi di mana observasi tersebut sudah diketahui kategori-kategorinya. Klasifikasi merupakan suatu contoh dari *supervised learning,* yaitu pembelajaran di mana *training set* dari observasi yang telah diidentifikasi dengan benar telah tersedia (Ethem, 2010).

Sementara itu, *classifier* adalah sebuah algoritma yang mengimplementasikan klasifikasi, khususnya secara konkret. Terdapat banyak *classifier* yang dapat digunakan untuk melakukan klasifikasi terhadap sebuah permasalahan. Untuk laporan ini, terdapat tiga buah *classifier* yang digunakan, yaitu *decision tree, neural network*, dan *Bayesian network.*

2.1.1. Decision tree

Decision tree merupakan metode ang memperkirakan fungsi target bernilai diskrit, di mana fungsi pembelajaran direpresentasikan oleh sebuah pohon keputusan. *Decision tree* mempelajari *decision rules* sederhana yang dapat diinferensi dari fitur-fitur data yang tersedia (Sulhan, n.d.).

Secara umum, *decision tree* cocok untuk digunakan pada permasalahan yang direpresentasikan dengan pasangan atribut-nilai serta mempunyai fungsi target yang bernilai diskrit. Kelebihan penggunaan *decision tree* antara lain mudah untuk dipahami dan divisualisasikan, tidak membutuhkan banyak persiapan data, serta *cost* relatif tidak besar.

Namun, *decision tree* memiliki beberapa kekurangan. Salah satu di antaranya adalah *decision tree* dapat menghasilkan *tree* yang terlalu kompleks sehingga tidak dapat melakukan generalisasi terhadap data dengan baik *(overfitting).* Masalah ini dapat dihindari dengan melakukan mekanisme seperti *pruning,* yaitu

menentukan jumlah sampel minimum pada sebuah *leaf node* atau menetapkan kedalaman maksimal dari *tree* yang dihasilkan.

Pada laporan ini, saya menggunakan J48 *decision tree* yang sudah di-*pruning*. J48 merupakan implementasi dari algoritma C4.5, di mana algoritma C4.5 merupakan suksesor dari algoritma ID3 (*Iterative Dichotomiser 3*).

2.1.2. Neural network

Neural network terinspirasi dari sistem pembelajaran biologis yang terbangun dari neuron yang saling terhubung dalam jaringan yang sangat kompleks. *Neural network* dibangun dari sekumpulan neuron (atau unit) yang saling terhubung. Setiap unit menerima sejumlah input baik itu input langsung maupun input dari unit lain dan menghasilkan sebuah nilai yang bisa juga menjadi input bagi unit lain.

Masing-masing unit pada *neural network* akan mengaplikasikan fungsi pada input tersebut sebelum kemudian dikirim ke layer lain. Pada umumnya, jaringan didefinisikan sebagai jaringan yang *feed-forward:* sebuah unit akan mengirim output yang dihasilkan ke layer lain, namun tidak ada umpan balik ke layer sebelumnya. Berat akan diaplikasikan kepada sinyal dari satu unit ke unit lainnya, di mana nilai berat inilah yang akan diatur pada setiap fase *training*. Inilah fase *learning* dari sebuah *neural network.*

Neural network membutuhkan waktu yang lama untuk belajar, namun setelah proses pembelajaran selesai, *neural network* dapat dengan cepat mengklasifikasikan observasi baru.

Pada laporan ini, saya menggunakan *classifier* MultilayerPerceptron pada Weka yang merupakan implementasi dari *neural network*.

2.1.3. Bayesian network

Bayesian network adalah model representasi grafik dari random variabel beserta dengan hubungan antara *random variable* tersebut dengan tanda panah (Pearl, 1998). Pada *Bayesian network,* terdapat CPD atau *Conditional Probability Distribution* yang merupakan nilai representasi dari probabilitas antar *random variable.*

Untuk proses klasifikasi, pada *Bayesian network* terbagi menjadi dua sub-proses, yaitu proses pembelajaran *(learning process)* dan proses pendugaan *(inference process).* Tujuan dari proses pembelajaran adalah untuk membentuk struktur dari *Bayesian network*, sementara tujuan dari proses pendugaan adalah klasifikasi instan.

2.2. Cross validation

Cross validation adalah salah satu teknik yang dapat digunakan untuk melakukan validasi terhadap keakuratan sebuah model yang dibangun berdasarkan *dataset* tertentu.

2.2.1. K-fold cross validation

Salah satu metode *cross validation* yang populer adalah *k-fold cross validation,* di mana *dataset* dibagi menjadi sejumlah k-buah partisi secara acak. Kemudian, dilakukan sejumlah k-kali eksperimen, di mana masing-masing eksperimen menggunakan data partisi ke-k sebagai *testing data* dan memanfaatkan sisa

partisi lainnya yang tidak digunakan sebagai *training data.* Selanjutnya, nilai rata-rata dari *error* pada semua k percobaan dihitung (Schneider, n.d.).

Keuntungan dari *k-fold cross validation* adalah cara pembagian *dataset* tidak begitu dipermasalahkan, karena setiap *data point* akan berada pada *testing data* sebanyak tepat satu kali dan berada pada *training data* sebanyak k-1 kali. Dengan *k-fold cross validation,* nilai varians dari hasil akan berkurang seiring dengan peningkatan nilai k. Kekurangan dari *k-fold cross validation* adalah algoritma harus dijalankan sebanyak k, sehingga untuk dapat menyelesaikan evaluasi, maka dibutuhkan waktu yang lebih lama dari yang seharusnya.

Untuk eksperimen 1 sampai dengan eksperimen 3 yang akan dilakukan selanjutnya, akan digunakan teknik *10-fold cross validation* yang fungsinya sudah dapat langsung digunakan pada Weka. Dengan teknik tersebut, *dataset* akan dibagi menjadi 10 partisi dan eksperimen akan dijalankan 10 kali dengan 10 *testing data* yang berbeda-beda.

2.2.2. Holdout method

Pada *holdout method, dataset* dibagi menjadi dua buah *set* yang berbeda, yaitu *training set* dan *testing set.* Model akan diminta untuk memprediksi nilai *output* untuk data yang ada pada *testing set*. Selanjutnya, seluruh *error* yang dihasilkan oleh model akan diakumulasikan sebagai nilai *mean absolute test set error* yang akan digunakan untuk mengevaluasi model.

Kelebihan dari *holdout method* adalah waktu yang dibutuhkan singkat. Namun, evaluasi yang dilakukan dapat memiliki nilai varians yang tinggi. Hal ini karena evaluasi sangat bergantung terhadap data yang ada pada *training set* dan *testing set* (Schneider, n.d.).

2.3. Evaluasi terhadap hasil analisis

Performa dari *classifier* yang digunakan diukur berdasarkan *accuracy, precision, recall, F-measure,* dan *ROC area* yang didefinisikan sebagai berikut:

2.3.1. Accuracy

Nilai keakuratan dari sebuah *classifier* merupakan persentase dari *instance* yang berhasil diklasifikasikan dengan benar oleh model. Nilai keakuratan didapatkan melalui perhitungan berikut:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \; x \; 100\%$$

2.3.2. Precision

Nilai *precision* dari sebuah kelas menggambarkan seberapa banyak *instance* yang diklasifikasikan sebagai suatu kelas tertentu sebenarnya memang merupakan kelas tersebut. Nilai *precision* didapatkan melalui perhitungan berikut:

$$Precision = \frac{TP}{TP+FP} \; x \; 100\%$$

2.3.3. Recall

Nilai *recall* dari sebuah kelas menggambarkan proporsi dari jumlah *instance* yang diklasifikasikan dengan tepat. Nilai *recall* didapatkan melalui perhitungan berikut:

$$Recall = \frac{TP}{TP+FN} \, x \, 100\%$$

2.3.4. F-measure

Nilai *F-measure* menggabungkan nilai *precision* dan nilai *recall* sebagai rata-rata harmonik. Nilai *F-measure* didapatkan melalui perhitungan berikut:

$$F-measure = \frac{2 \, x \, recall \, x \, precision}{recall + precision} \, x \, 100\%$$

2.3.5. ROC area

Kurva ROC merupakan kurva yang membandingkan antara *true positive rate* dan *false positive rate* yang dihasilkan oleh model. Semakin kurva mendekati sisi kiri dan sisi atas dari grafik, maka dapat disimpulkan bahwa performa dari *classifier* tersebut baik. Oleh karena itu, semakin nilai dari luas di bawah kurva ROC mendekati 1, maka performa *classifier* semakin baik. Sebaliknya, apabila kurva mendekati garis diagonal dari grafik, maka *classifier* cenderung membuat prediksi yang mendekati *random guessing.* Hal ini digambarkan dengan nilai dari luas di bawah kurva ROC yang mendekati 0.5.

Untuk menentukan *classifier* terbaik, nilai-nilai seperti *precision, recall, F-measure,* dan *ROC area* juga dijadikan pertimbangan untuk menghindari bias yang dapat ditimbulkan apabila *classifier* hanya dinilai dari tingkat akurasi saja ("*Evaluate model performance in Machine Learning",* 2016).

F-measure merupakan nilai yang penting untuk mengevaluasi *classifier* karena *classifier* yang ideal, khususnya untuk identifikasi bahasa, memiliki sedikit *false positives* karena *false positives* dapat mengakibatkan semakin banyaknya dokumen yang harus di-*retrieve* karena kesalahan dalam mengklasifikasikan bahasa. Selain itu dibutuhkan juga jumlah *false negatives* yang minimal karena kesalahan dalam mengklasifikasikan bahasa dapat mengakibatkan terlewatnya dokumen-dokumen yang sebenarnya terklasifikasi dalam suatu bahasa.

3. Representasi Masalah dan Pengaturan Fitur

3.1. Representasi masalah dalam machine learning

Seluruh algoritma *machine learning* memiliki tiga komponen, yaitu:

- Representasi*:* bagaimana algoritma merepresentasikan *knowledge* yang dimiliki. Contohnya adalah dengan *decision tree, neural network, support vector machine,* himpunan *rules,* dan lain sebagainya.
- Evaluasi: cara untuk mengevaluasi hipotesis. Contohnya adalah akurasi, *precision* dan *recall, squared error, likelihood, posterior probability, cost, margin,* dan lain sebagainya.

- Optimisasi: *search process* seperti *combinatorial optimization, convex optimization,* dan *constrained optimization*.

Pada *machine learning,* sebuah *classifier* harus direpresentasikan sebagai bahasa formal yang dapat diolah oleh komputer (Domingos, 2010).

3.2. Ekstraksi kalimat dari berkas XML

Pada eksperimen ini, pengambilan kalimat dari *file* XML dilakukan dengan sebuah *script* Python xml2txt.py yang dibuat oleh penulis. *Script* tersebut menggunakan ElementTree XML API untuk mengolah *file* XML yang diberikan serta *library* Natural Language Toolkit (NLTK) untuk pengolahan bahasa yang akan dijelaskan lebih lanjut. Berikut merupakan langkah-langkah yang dilakukan oleh xml2txt.py:

1. ***Append <ROOT> to file:*** Karena pada *file* Combination.xml yang diberikan tidak terdapat sebuah *common node* yang melingkupi seluruh isi dari file XML tersebut, maka *file* Combination.xml tidak dapat langsung diproses. Oleh karena itu, hal pertama yang dilakukan oleh xml2txt.py adalah menambahkan *tag* <ROOT> pada *file* Combination.xml.
2. **Mengolah *tags:*** Selanjutnya, *script* akan mengekstrak nilai maupun konten yang dibutuhkan berdasarkan *tags* XML yang terdapat pada Combination.xml dengan menggunakan ElementTree XML API. Seluruh nilai maupun konten yang dibutuhkan disimpan dalam variabel masing-masing agar dapat diakses kemudian.
3. ***Append <ROOT> to file:*** Karena pada *file* Combination.xml yang diberikan tidak terdapat sebuah *common node* yang melingkupi seluruh isi dari file XML tersebut, maka *file* Combination.xml tidak dapat langsung diproses. Oleh karena itu, hal pertama yang dilakukan oleh xml2txt.py adalah menambahkan *tag* <ROOT> pada *file* Combination.xml.
4. **Mengolah *tags:*** Selanjutnya, *script* akan mengekstrak nilai maupun konten yang dibutuhkan berdasarkan *tags* XML yang terdapat pada Combination.xml dengan menggunakan ElementTree XML API. Seluruh nilai maupun konten yang dibutuhkan disimpan dalam variabel masing-masing agar dapat diakses kemudian.
5. **Memisahkan konten menjadi kalimat-kalimat:** Untuk memisahkan konten menjadi kalimat-kalimat, *script* xml2txt.py menggunakan *library* NLTK dengan *module* Punkt. Hal ini agar *script* dapat memisahkan kalimat-kalimat yang ada dengan lebih akurat dengan memperhatikan hal-hal seperti kalimat langsung, singkatan, dan lain-lain. Namun perlu dicatat bahwa dengan menggunakan NLTK masih terdapat pemisahan kalimat yang kurang tepat. Untuk setiap bahasa digunakan modul yang berbeda sesuai dengan bahasa masing-masing, meskipun untuk konten berbahasa Indonesia digunakan *module* untuk bahasa Inggris karena *module* untuk bahasa Indonesia tidak tersedia.
6. **Mencetak *output* ke *file* Sentences.txt:** Seluruh informasi yang telah berhasil diekstrak kemudian dicetak ke *file* Sentences.txt sesuai dengan format yang telah ditentukan pada soal.

3.3. Persiapan data masukan untuk Weka

Setelah berkas Sentences.txt dihasilkan, maka akan dilakukan proses *feature extraction* untuk mengambil fitur-fitur yang terdapat pada kalimat dengan menggunakan program FeatureGenerator.java yang

telah disediakan di Scele. Pada program tersebut, kita dapat memilih representasi yang diinginkan, seperti representasi berdasarkan fitur N-gram (N=1, 2, 3, 4) dan fitur boolean atau frekuensi.

Sebagai catatan, terdapat beberapa bagian dari program FeatureGenerator.java yang saya modifikasi, yaitu pada baris ke-536 yang semula adalah:

```
this.isi = myIsi.replaceAll("[.,?!;'\"!@#$%^&*()-=_+\\[\\]{}:<>/|\\\\]","").replaceAll("[^\\x00-\\x7F]","");
```

Menjadi:

```
this.isi = myIsi.replaceAll("[.,?!;'\"!@#$%^&*()-=_+\\[\\]{}:<>/|\\\\]","").replaceAll(" {2,}"," ");
```

Bagian `replaceAll("[^\\x00-\\x7F]","");` dihapus karena kode tersebut menghilangkan seluruh karakter yang bukan merupakan karakter ASCII. Hal ini mengakibatkan *special characters* seperti huruf-huruf asing beraksen menjadi hilang karena karakter-karakter tersebut bukanlah merupakan karakter ASCII. Hal ini menjadi *concern* karena sebagai contoh, pada Bahasa Prancis, terdapat kata yang hanya terdiri dari satu huruf saja seperti **à**. Apabila kita menghapus karakter tersebut maka kita akan menghapus sebuah kata, di mana hilangnya suatu kata dapat berpengaruh pada pembuatan model.

Modifikasi selanjutnya adalah penambahan bagian `replaceAll(" {2,}"," ");`, karena pada program asli, apabila dijalankan akan menghasilkan beberapa spasi yang *redundant* pada berkas yang dihasilkan.

Setelah berkas Feature.txt dihasilkan, agar dapat melakukan klasifikasi dengan menggunakan Weka, berkas perlu dipersiapkan sehingga berkas tersebut dapat diolah oleh Weka. Salah satu data format yang dapat menjadi masukan untuk Weka adalah ARFF (*Attribute-Relation File Format*). Namun, karena fitur-fitur hasil ekstraksi yang terdapat pada Feature.txt bukanlah format yang dapat diterima oleh Weka, maka perlu dilakukan *pre-processing* sehingga berkas tersebut dapat diterima Weka.

ARFF memiliki dua bagian utama, yaitu Header yang kemudian diikuti oleh Data. Bagian Header dari berkas ARFF sendiri terdiri dari nama dari relasi (yang diawali dengan @RELATION) serta daftar atribut dan tipe data dari masing-masing atribut tersebut (yang diawali dengan @ATTRIBUTE).

Untuk *pre-processing,* saya membuat *script* Python yang bernama txt2arff.py. Secara garis besar, *script* tersebut berfungsi untuk membuat berkas yang sesuai dengan format ARFF. Dalam eksperimen ini, berkas yang menjadi *input* adalah berkas yang dihasilkan oleh program FeatureGenerator.java.

Perlu diketahui bahwa hasil *output* dari FeatureGenerator.java sendiri kurang lebih sudah berbentuk hampir sama dengan format bagian Data (yang diawali dengan @DATA) dari sebuah berkas ARFF. Sehingga, *script* txt2arff.py hanya perlu membuat bagian Header dari berkas ARFF tersebut.

Pada pencetakan informasi @RELATION di berkas ARFF, *script* akan mencetak nama relasi sesuai dengan nama berkas yang menjadi *input* dari *script* txt2arff.py.

Pada pencetakan informasi @ATTRIBUTE, *script* akan menyimpan seluruh atribut yang diambil dari baris pertama Feature.txt, kemudian mencetaknya sesuai dengan format pada berkas ARFF. Untuk menyederhanakan kalkulasi, mempertimbangkan bahwa *dataset* yang diolah cukup besar, maka pada *script*

txt2arff.py seluruh informasi terkait *data type* dari masing-masing atribut telah di-*hardcode.* Untuk seluruh atribut sudah dipastikan *data type*-nya NUMERIC dan untuk *data type* dari kelas (yang pada *script* bernama language_class) merupakan NOMINAL dengan nilai-nilai yaitu: {Indonesia,Inggris,Spanyol,Perancis}.

4. Hasil uji coba dan analisis

4.1. Eksperimen 1

Pada eksperimen 1, uji coba *training* dilakukan dengan menggunakan tiga jenis *classifier,* yaitu *decision tree, neural network,* dan *Bayesian network.* Ekstraksi fitur yang digunakan adalah Boolean 1-gram.

4.1.1. Decision tree

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     Feature-Boolean-1-gram
Instances:    8117
Attributes:   493
              [list of attributes omitted]
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

fue <= 0
|   lo <= 0
|   |   sus <= 0
|   |   |   desde <= 0
|   |   |   |   también <= 0
|   |   |   |   |   este <= 0
|   |   |   |   |   |   parte <= 0
|   |   |   |   |   |   |   españa <= 0
|   |   |   |   |   |   |   |   sobre <= 0
|   |   |   |   |   |   |   |   |   hasta <= 0
|   |   |   |   |   |   |   |   |   |   pero <= 0
|   |   |   |   |   |   |   |   |   |   |   bronce <= 0
|   |   |   |   |   |   |   |   |   |   |   |   itu <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   atau <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   jawa <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   juga <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   tidak <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   merupakan <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   menjadi <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   ia <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   kerajaan <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   seperti <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   lebih <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   masa <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   sebuah <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   satu <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   memiliki <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   aux <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   avec <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   mais <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   ou <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   sa <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   ne <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   deux <= 0
```

```
|   |   |   |   |   |   |   |   |   hasta > 0: Spanyol (30.0)
|   |   |   |   |   |   |   |   sobre > 0: Spanyol (34.0)
|   |   |   |   |   |   |   españa > 0: Spanyol (36.0)
|   |   |   |   |   |   |   parte > 0: Spanyol (46.0)
|   |   |   |   |   este > 0: Spanyol (47.0)
|   |   |   |   también > 0: Spanyol (53.0)
|   |   |   desde > 0: Spanyol (61.0)
|   |   sus > 0: Spanyol (73.0)
|   lo > 0: Spanyol (81.0)
fue > 0: Spanyol (96.0)

Number of Leaves  :     60

Size of the tree :      119


Time taken to build model: 125.26 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        6108               75.2495 %
Incorrectly Classified Instances      2009               24.7505 %
Kappa statistic                          0.6089
Mean absolute error                      0.1857
Root mean squared error                  0.3074
Relative absolute error                 53.4128 %
Root relative squared error             73.7305 %
Total Number of Instances             8117

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.992    0.437    0.647      0.992    0.783      0.594    0.778     0.648     Inggris
                 0.578    0.000    0.999      0.578    0.732      0.726    0.846     0.730     Indonesia
                 0.610    0.006    0.967      0.610    0.749      0.724    0.846     0.764     Perancis
                 0.458    0.001    0.984      0.458    0.625      0.640    0.800     0.615     Spanyol
Weighted Avg.    0.752    0.197    0.832      0.752    0.743      0.654    0.809     0.684

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 3601    0   30    0 |    a = Inggris
  628  861    0    0 |    b = Indonesia
  689    0 1094    9 |    c = Perancis
  645    1    7  552 |    d = Spanyol
```

Fig. 1. (a) *Classification output* untuk eksperimen 1 dengan menggunakan *decision tree* sebagai *classifier*; (b) lanjutan

4.1.2. Neural network

```
=== Run information ===

Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 100 -V 0 -S 0 -E 20 -H 5
Relation:     Feature-Boolean-1-gram
Instances:    8117
Attributes:   493
              [list of attributes omitted]
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
    Inputs    Weights
    Threshold    -6.6849133895772885
    Node 4    2.260576944968363
    Node 5    -2.6803121514257158
    Node 6    2.6648918401174284
    Node 7    3.2278564977183444
    Node 8    3.368083187261597
Sigmoid Node 1
    Inputs    Weights
    Threshold    -6.956927633240049
    Node 4    3.202890049628602
    Node 5    6.633519443592397
    Node 6    -4.828553782528523
    Node 7    1.9462840901571317
    Node 8    -6.217218676986156
Sigmoid Node 2
    Inputs    Weights
    Threshold    4.257901010337733
    Node 4    -2.737049631287979
    Node 5    -5.016967194937494
    Node 6    -2.8700246742432056
    Node 7    -3.3048374897771207
    Node 8    -2.51472638081858
Sigmoid Node 3
    Inputs    Weights
    Threshold    -4.995311360557158
    Node 4    -2.6326431323018675
    Node 5    4.1459285037199045
    Node 6    3.1874170860280238
    Node 7    -2.4746895395371857
    Node 8    4.053824759728538
Sigmoid Node 4
    Inputs    Weights
    Threshold    0.8687845239641852
    Attrib some    5.298662125087003
    Attrib when    6.028131084657225
```

```
Time taken to build model: 134.56 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        7135               87.9019 %
Incorrectly Classified Instances       982               12.0981 %
Kappa statistic                          0.824
Mean absolute error                      0.0768
Root mean squared error                  0.214
Relative absolute error                 22.0873 %
Root relative squared error             51.3259 %
Total Number of Instances             8117

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.942    0.093    0.892      0.942    0.916      0.846    0.964     0.944     Inggris
                 0.844    0.026    0.880      0.844    0.861      0.831    0.965     0.925     Indonesia
                 0.870    0.050    0.831      0.870    0.850      0.807    0.947     0.922     Perancis
                 0.746    0.011    0.919      0.746    0.824      0.802    0.855     0.827     Spanyol
Weighted Avg.    0.879    0.059    0.880      0.879    0.878      0.828    0.944     0.919

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 3421   57  114   39 |    a = Inggris
   92 1256  124   17 |    b = Indonesia
  147   63 1559   23 |    c = Perancis
  177   51   78  899 |    d = Spanyol
```

Fig. 2. (a) *Classification output* untuk eksperimen 2 dengan menggunakan *decision tree* sebagai *classifier*; (b) lanjutan

4.1.3. Bayesian Network

```
=== Run information ===

Scheme:         weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2
-- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

Relation:       Feature-Boolean-1-gram-weka.filters.unsupervised.attribute.Remove-R1

Instances:      8117

Attributes:   493

                [list of attributes omitted]

Test mode:      10-fold cross-validation

Time taken to build model: 1.84 seconds

=== Stratified cross-validation ===
=== Summary ===
```

```
Correctly Classified Instances          7504               92.4479 %
Incorrectly Classified Instances        613                7.5521 %
Kappa statistic                         0.8932
Mean absolute error                     0.0442
Root mean squared error                 0.1446
Relative absolute error                 12.7134 %
Root relative squared error             34.6829 %
Total Number of Instances               8117

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area  Class
               0.925    0.005    0.994    0.925    0.958    0.928    0.995    0.994    Inggris
               0.923    0.001    0.997    0.923    0.958    0.951    0.996    0.985    Indonesia
               0.881    0.003    0.990    0.881    0.932    0.917    0.993    0.978    Perancis
               0.991    0.083    0.676    0.991    0.804    0.783    0.989    0.947    Spanyol
Weighted Avg.  0.924    0.015    0.946    0.924    0.930    0.908    0.994    0.982

=== Confusion Matrix ===

       a       b       c       d   <-- classified as
 3357   3   14  257 |   a = Inggris
       3 1374  0  112 |        b = Indonesia
   10   0 1579  203 |   c = Perancis
       8       1       2 1194 |        d = Spanyol
```

4.1.4. Kesimpulan untuk eksperimen 1

Table 1. Perbandingan performa antara *classifier decision tree, neural network,* dan *Bayesian network*

Metode	*Accuracy (TP Rate)*	*Precision*	*Recall*	*F-Measure*	*ROC Area*	*Waktu (dalam sekon)*
Decision tree	0.752	0.832	0.752	0.743	0.809	125.26
Neural network	0.879	0.880	0.879	0.878	0.944	134.56
Bayesian network	0.924	0.946	0.924	0.930	0.994	1.84

Dari tabel diatas, dapat dilihat bahwa secara keseluruhan performa *decision tree* dan *neural network* berada di bawah *Bayesian network.* Selain itu, waktu yang dibutuhkan untuk membuat model pada *Bayesian network* juga jauh lebih sedikit. Sehingga, dapat disimpulkan bahwa *Bayesian network* merupakan *classifier* terbaik di antara ketiga *classifier* yang telah digunakan.

4.2. Eksperimen 2

Pada eksperimen 2, uji coba *training* dilakukan dengan menggunakan *classifier* berjenis *Bayesian network.* Ekstraksi fitur yang digunakan adalah frekuensi 1-gram.

4.2.1. Frekuensi 1-gram

```
=== Run information ===

Scheme:         weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2
-- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5
Relation:       Feature-Frekuensi-1-gram-weka.filters.unsupervised.attribute.Remove-R1
Instances:      8117
Attributes:   493
                [list of attributes omitted]
Test mode:      10-fold cross-validation
Time taken to build model: 1.92 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          7504            92.4479 %
Incorrectly Classified Instances        613             7.5521 %
Kappa statistic                         0.8932
Mean absolute error                     0.0442
Root mean squared error                 0.1446
Relative absolute error                 12.7133 %
Root relative squared error             34.6828 %
Total Number of Instances               8117

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area  Class
                0.925    0.005    0.994    0.925    0.958    0.928    0.995    0.994    Inggris
                0.923    0.001    0.997    0.923    0.958    0.951    0.996    0.985    Indonesia
                0.881    0.003    0.990    0.881    0.932    0.917    0.993    0.978    Perancis
                0.991    0.083    0.676    0.991    0.804    0.783    0.989    0.947    Spanyol
Weighted Avg.   0.924    0.015    0.946    0.924    0.930    0.908    0.994    0.982

=== Confusion Matrix ===

        a       b       c       d   <-- classified as
 3357   3   14  257 |   a = Inggris
        3 1374  0  112 |        b = Indonesia
   10   0 1579  203 |   c = Perancis
        8       1       2 1194 |        d = Spanyol
```

4.2.2. Kesimpulan untuk eksperimen 2

Table 2. Perbandingan performa antara ekstraksi fitur boolean dan frekuensi

Fitur	*Accuracy (TP Rate)*	*Precision*	*Recall*	*F-Measure*	*ROC Area*	*Waktu (dalam sekon)*
Boolean	0.924	0.946	0.924	0.930	0.994	1.84
Frekuensi	0.924	0.946	0.924	0.930	0.994	1.84

Berdasarkan eksperimen 1 dan eksperimen 2, didapatkan bahwa proses klasifikasi dengan menggunakan baik ekstraksi fitur boolean maupun frekuensi pada *Bayesian network* dengan menggunakan nilai n = 1 untuk n-gram akan menghasilkan hasil yang sama.

Untuk eksperimen-eksperimen selanjutnya akan digunakan *dataset* hasil ekstraksi fitur boolean 1-gram.

4.3. Eksperimen 3

Pada eksperimen 3, uji coba *training* dilakukan dengan menggunakan *classifier* terbaik dari eksperimen 1 yaitu *Bayesian network* dan jenis fitur boolean. Namun pada eksperimen 3 akan dilakukan uji coba *training* dengan berbagai variasi untuk n-gram, yaitu 1-gram, 2-gram, 3-gram, dan 4-gram. Karena eksperimen untuk Boolean 1-gram dengan *Bayesian network* telah dilaksanakan di eksperimen 1, maka untuk eksperimen 3 hanya dilakukan eksperimen dengan 2-gram, 3-gram, dan 4-gram.

4.3.1. 2-gram

```
=== Run information ===

Scheme:         weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2
-- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5
Relation:       Feature-Boolean-2-gram-weka.filters.unsupervised.attribute.Remove-R1
Instances:      8117
Attributes:   101
                [list of attributes omitted]
Test mode:      10-fold cross-validation

Time taken to build model: 12.07 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          5713            70.3831 %
Incorrectly Classified Instances        2404            29.6169 %
Kappa statistic                         0.5894
Mean absolute error                     0.1626
Root mean squared error                 0.2884
```

```
Relative absolute error                 46.7662 %
Root relative squared error             69.1558 %
Total Number of Instances               8117

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area  Class
               0.687    0.001    0.999    0.687    0.814    0.739    0.916    0.914    Inggris
               0.999    0.337    0.399    0.999    0.571    0.514    0.854    0.535    Indonesia
               0.574    0.013    0.927    0.574    0.709    0.678    0.896    0.782    Perancis
               0.650    0.001    0.995    0.650    0.786    0.780    0.925    0.818    Spanyol
Weighted Avg.  0.714    0.065    0.872    0.714    0.742    0.690    0.902    0.801

=== Confusion Matrix ===

       a       b       c       d   <-- classified as
 2494 1137     0       0 |     a = Inggris
       2 1487  0       0 |     b = Indonesia
       0  759 1029     4 |     c = Perancis
       1  340   81  783 |     d = Spanyol
```

4.3.2. 3-gram

```
=== Run information ===

Scheme:         weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2
-- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5
Relation:       Feature-Boolean-3-gram-weka.filters.unsupervised.attribute.Remove-R1
Instances:      8117
Attributes:   101
               [list of attributes omitted]
Test mode:      10-fold cross-validation

Time taken to build model: 0.19 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          4209            51.8541 %
Incorrectly Classified Instances        3908            48.1459 %
Kappa statistic                         0.1569
Mean absolute error                     0.3102
Root mean squared error                 0.3936
Relative absolute error                 89.2009 %
Root relative squared error             94.3936 %
Total Number of Instances               8117

=== Detailed Accuracy By Class ===
```

```
                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area  Class
                1.000   0.871   0.482   1.000   0.650   0.249   0.586   0.532   Inggris
                0.028   0.000   1.000   0.028   0.054   0.150   0.566   0.230   Indonesia
                0.170   0.000   1.000   0.170   0.290   0.371   0.612   0.402   Perancis
                0.193   0.000   1.000   0.193   0.324   0.412   0.621   0.352   Spanyol
Weighted Avg.   0.519   0.390   0.768   0.519   0.413   0.282   0.593   0.421

=== Confusion Matrix ===

       a       b       c       d   <-- classified as
 3631   0       0       0 |     a = Inggris
 1448   41      0       0 |     b = Indonesia
 1488   0  304  0 |     c = Perancis
  972   0       0  233 |        d = Spanyol
```

4.3.3. 4-gram

```
=== Run information ===

Scheme:         weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2
-- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5
Relation:       Feature-Boolean-4-gram-weka.filters.unsupervised.attribute.Remove-R1
Instances:      8117
Attributes:   101
                [list of attributes omitted]
Test mode:      10-fold cross-validation

Time taken to build model: 0.2 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          3763            46.3595 %
Incorrectly Classified Instances        4354            53.6405 %
Kappa statistic                         0.0365
Mean absolute error                     0.3403
Root mean squared error                 0.4126
Relative absolute error                 97.8563 %
Root relative squared error             98.9437 %
Total Number of Instances               8117

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area  Class
                1.000   0.971   0.455   1.000   0.625   0.116   0.515   0.455   Inggris
                0.000   0.000   0.000   0.000   0.000   0.000   0.510   0.186   Indonesia
```

```
                0.049   0.000   1.000   0.049   0.094   0.197   0.526   0.273   Perancis
                0.037   0.000   1.000   0.037   0.070   0.177   0.527   0.190   Spanyol
Weighted Avg.   0.464   0.434   0.573   0.464   0.311   0.121   0.518   0.326

=== Confusion Matrix ===

       a       b       c       d   <-- classified as
 3631   0       0       0 |     a = Inggris
 1489   0       0       0 |     b = Indonesia
 1704   0   88  0 |     c = Perancis
 1161   0       0   44 |        d = Spanyol
```

4.3.4. Kesimpulan eksperimen 3

Table 3. Perbandingan performa antara ekstraksi fitur 1-gram, 2-gram, 3-gram, dan 4-gram

n-gram	*Accuracy (TP Rate)*	*Precision*	*Recall*	*F-Measure*	*ROC Area*	*Waktu (dalam sekon)*
1-gram	0.924	0.946	0.924	0.930	0.994	1.84
2-gram	0.714	0.872	0.714	0.725	0.902	0.23
3-gram	0.519	0.768	0.519	0.409	0.593	0.19
4-gram	0.464	0.573	0.464	0.311	0.518	0.2

Berdasarkan eksperimen 1 dan 3, didapat bahwa secara keseluruhan, performa dari *classifier Bayesian network* yang menggunakan ekstraksi fitur boolean 1-gram lebih baik apabila dibandingkan dengan variasi n-gram yang lainnya (2-gram, 3-gram, 4-gram). Dapat dilihat bahwa dalam segala aspek, semakin meningkat nilai n pada n-gram, maka akan semakin menurun baik nilai *accuracy, precision, recall, f-measure,* maupun *ROC area.* Penurunan yang cukup drastis terdapat antara ekstraksi fitur dengan 2-gram dan 3-gram. Oleh karena itu, dapat disimpulkan bahwa variasi n-gram dengan nilai n yaitu 1 merupakan n-gram terbaik untuk digunakan pada *classifier* berjenis *Bayesian network* dengan ekstraksi fitur berjenis boolean.

Kemungkinan besar nilai n = 1 dapat memberikan hasil yang terbaik adalah karena *dataset* yang dijadikan masukan tidak cukup besar. Ketika *training set* untuk sebuah *classifier* terlalu kecil, besar kemungkinan bahwa ada banyak variasi dari n-gram (seperti 2-gram, 3-gram, maupun 4-gram) yang tidak terlihat di *training set* namun muncul di *testing set.*

4.4. Eksperimen 4

Pada eksperimen 4, akan dilakukan *uji coba training* dengan menggunakan *feature selection* pada Weka. *Classifier* yang digunakan adalah *Bayesian network* dengan ekstraksi fitur boolean 1-gram menggunakan *feature selection.*

Proses *feature selection* digunakan adalah *feature selection* dengan menghitung *information gain* (atau entropi) dari masing-masing atribut untuk variabel *output.* Nilai *information gain* berkisar antara 0 (tidak ada informasi) sampai 1 (informasi maksimum) (Mitchell, 1997). Atribut-atribut yang memberikan lebih banyak informasi akan memiliki nilai *information gain* yang lebih tinggi dan dapat dipilih, sementara

atribut-atribut yang tidak memberikan banyak informasi akan memiliki nilai yang lebih rendah dan dapat dihilangkan.

Pada Weka, proses *feature selection* dengan menggunakan *information gain* dilakukan dengan menggunakan *InfoGainAttributeEval Attribute Evaluator* dengan *Ranker Search Method*. Parameter yang digunakan adalah sebagai berikut:

- numToSelect (jumlah atribut yang digunakan): 246
  Berikut merupakan daftar atribut yang terpilih:

0.04351	4 itu	0.02042	94 lebih	0.01734	143 barat
0.04193	9 juga	0.02042	105 ada	0.01707	117 ligne
0.04193	6 atau	0.02042	119	0.01706	157 han
0.0413	10 jawa	memiliki		0.01673	182 sebuah
0.03753	13 avec	0.02036	84 ont	0.01652	140 même
0.03753	5 aux	0.02036	79 ils	0.01642	180 sejak
0.03363	20 mais	0.02036	93 ainsi	0.01625	133 leurs
0.03345	33 menjadi	0.02029	12 been	0.01611	201 salah
0.03314	44 fue	0.02011	128 dapat	0.01611	156
0.03174	46 lo	0.02011	112 bahwa	sumatera	
0.03168	19 ou	0.02011	125 dikenal	0.01611	169 sekitar
0.03158	36 tidak	0.01988	174 gran	0.01604	26 had
0.03044	22 ne	0.01981	86 comme	0.01604	27 many
0.02971	50	0.01981	87 guerre	0.01598	153 premier
merupakan		0.0198	135 satu	0.01598	163 fait
0.02961	21 ce	0.01955	14 first	0.01589	30 its
0.0294	41 seperti	0.01919	190 esta	0.01589	34 then
0.02928	61 sus	0.01899	103 leur	0.01581	89 wayang
0.02788	73 desde	0.01899	107 après	0.01581	206 setelah
0.02753	57 ia	0.01899	110 ces	0.01581	161 sunda
0.02752	37 sa	0.01888	147 telah	0.01575	35 time
0.02719	80 este	0.01857	127	0.01573	199 hierro
0.02691	16 kerajaan	beberapa		0.01573	257 está
0.02687	25 pas	0.01853	8 she	0.0157	166 avions
0.02649	74 hasta	0.01853	17 while	0.0156	31 two
0.02381	52 abad	0.018	139 masa	0.01546	38 between
0.02338	60 ses	0.01796	56 batik	0.01543	165 sous
0.02335	109 parte	0.01796	118 nama	0.01543	175 alors
0.0232	88 karena	0.01794	15 them	0.01539	262 aunque
0.02289	82 saat	0.01789	115 roi	0.01531	39 these
0.02283	64 boeing	0.01779	18 most	0.01516	158 années
0.02266	122 también	0.01765	141	0.01512	179 ha
0.02266	100 años	tersebut		0.01504	253 bronce
0.02205	1 some	0.01762	114 elle	0.01502	42 there
0.02201	76 cette	0.01746	214 pero	0.01501	244 forma
0.02201	67 deux	0.01746	221 siglo	0.01489	71 bahasa
0.02146	3 used	0.01736	24 after	0.01489	152 où
0.02132	2 when	0.01734	137 été	0.01487	40 may
0.02127	154 sobre	0.01734	148 bagian	0.01487	23 process
0.02092	136 españa	0.01734	101 belanda	0.0147	236 cobre

0.01461 160 dune
0.0146 28 all
0.01458 193 wilayah
0.01458 187 raja
0.01445 255 dos
0.01442 54 service
0.01435 303 muy
0.01434 198 aussi
0.01428 247 antara
0.01428 220 yaitu
0.01407 171 siècle
0.01407 194 tout
0.01401 325 otros
0.01401 290 año
0.014 43 where
0.014 53 example
0.014 51 during
0.01397 195 orang
0.01397 231 hingga
0.01397 229 sudah
0.01397 238 kemudian
0.01397 248 masih
0.01386 203 daerah
0.01383 72 france
0.01379 204 compagnies
0.01379 196 fut
0.01368 29 use
0.01366 155 gunung
0.01366 240 bentuk
0.01347 304 sin
0.01332 327 ser
0.01312 49 information
0.01305 307 namun
0.01305 249 seorang
0.01305 205 danau
0.01298 223 pouvoir
0.01283 58 only
0.01275 296 banyak
0.01275 312 hanya
0.01275 270 lain
0.01275 285 sangat
0.01275 233 disebut
0.01275 263 serta
0.01275 282 sampai
0.01271 243 très
0.0127 102 partir
0.01263 368 arqueológico
0.01244 318 sekarang
0.01243 202 dun
0.01243 269 cest
0.01229 208 museo
0.01229 328 especie
0.01229 355 durante
0.01229 334 restos
0.01216 265 noblesse
0.01214 185 seni
0.01214 295 dunia
0.01214 291 selatan
0.01214 329 pertama
0.01214 343 menggunakan
0.01214 345 nusantara
0.01211 32 music
0.01211 63 about
0.01203 200 dont
0.01194 387 europa
0.01192 11 new
0.01189 273 marché
0.01189 225 contre
0.01183 335 timur
0.01183 346 akan
0.01183 370 sehingga
0.01182 77 however
0.01167 70 system
0.01162 271 réacteurs
0.01162 288 révolution
0.0116 373 edad
0.01153 81 out
0.01152 341 tetapi
0.01145 241 douglas
0.01138 69 each
0.01135 254 société
0.01135 309 cependant
0.01126 381 donde
0.01126 446 puede
0.01126 455 así
0.01124 83 often
0.01124 75 using
0.01122 357 pulau
0.01122 313 mereka
0.01109 65 universities
0.01107 302 airbus
0.01107 314 septembre
0.01091 350 baru
0.0108 97 up
0.01066 98 early
0.01066 66 text
0.01061 410 membuat
0.01061 146 tari
0.01061 404 awal
0.01061 399 berbagai
0.01057 407 nacional
0.01057 447 fueron
0.01053 336 première
0.01053 338 depuis
0.01053 363 lui
0.01051 78 human
0.01051 59 summarization
0.01046 245 fin
0.01037 332 mayor
0.01031 424 populer
0.01031 384 dua
0.01026 305 lors
0.01026 361 tous
0.01026 337 trois
0.01023 479 todo
0.01023 375 territorio
0.0101 48 university
0.01008 90 century
0.01008 85 will
0.01008 47 coffee
0.01 445 secara
0.00999 364 mis
0.00993 91 would
0.00993 106 before
0.00992 7 social
0.00979 120 through
0.00977 276 dc
0.00972 400 également

```
 0.00972        383                 0.00972        372 être
réaction                           0.0097         433 berasal
```

4.4.1. Ekstraksi fitur dengan feature selection

```
=== Run information ===

Scheme:         weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2
-- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

Relation:
Feature-Boolean-1-gram-weka.filters.unsupervised.attribute.Remove-R1-weka.filters.supervised.attr
ibute.AttributeSelection-Eweka.attributeSelection.InfoGainAttributeEval-Sweka.attributeSelection.
Ranker -T 0.0 -N 246

Instances:      8117

Attributes:   247

                [list of attributes omitted]

Test mode:      10-fold cross-validation

Time taken to build model: 1.61 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances          7405            91.2283 %

Incorrectly Classified Instances        712             8.7717 %

Kappa statistic                         0.8697

Mean absolute error                     0.0622

Root mean squared error                 0.1814

Relative absolute error                 17.8998 %

Root relative squared error             43.4948 %

Total Number of Instances               8117

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area  Class

                0.997    0.154    0.839    0.997    0.911    0.839    0.978    0.971    Inggris

                0.905    0.000    0.999    0.905    0.950    0.941    0.989    0.967    Indonesia
```

```
                0.852   0.002   0.992   0.852   0.917   0.899   0.981   0.953   Perancis
                0.756   0.001   0.993   0.756   0.859   0.848   0.972   0.898   Spanyol
Weighted Avg.   0.912   0.070   0.925   0.912   0.912   0.873   0.980   0.955


=== Confusion Matrix ===

a       b       c       d   <-- classified as
 3619   0   10  2 |     a = Inggris
  140 1348      0       1 |     b = Indonesia
  262   0 1527  3 |     c = Perancis
  290   1       3  911 |        d = Spanyol
```

4.4.2. Kesimpulan untuk eksperimen 4

Table 4. Perbandingan performa antara ekstraksi fitur tanpa *feature selection* dan dengan *feature selection*

Feature selection	*Accuracy (TP Rate)*	*Precision*	*Recall*	*F-Measure*	*ROC Area*	*Waktu (dalam sekon)*
Tanpa *feature selection*	0.924	0.946	0.924	0.930	0.994	1.84
Dengan *feature selection*	0.912	0.925	0.912	0.912	0.980	1.61

Berdasarkan eksperimen 4, didapatkan bahwa secara keseluruhan, klasifikasi dengan menggunakan *classifier Bayesian network* dengan ekstraksi fitur boolean 1-gram tanpa menggunakan *feature selection* memiliki performa yang sedikit lebih baik dibandingkan dengan menggunakan *feature selection* dengan metode *information gain,* meskipun waktu yang dibutuhkan untuk pembuatan model dengan menggunakan *feature selection* lebih sedikit dibandingkan dengan tidak menggunakan *feature selection.*

Perlu dicatat bahwa hasil eksperimen tanpa menggunakan *feature selection* dan dengan menggunakan *feature selection* tidak memiliki perbedaan yang signifikan. Namun, dengan *feature selection* pembuatan model dapat berlangsung dengan lebih cepat.

4.5. Eksperimen 5

Pada eksperimen 5, dilakukan *uji coba testing* dengan menggunakan teks yang telah penulis kumpulkan sebelumnya. Teks dikumpulkan secara acak dan berasal dari berbagai sumber seperti situs berita, *blog,* artikel, dan lain-lain. Seluruh teks yang dikumpulkan tidak ada yang membahas topik yang sama. Perlu

dicatat bahwa beberapa teks memiliki kombinasi bahasa (misalnya, teks yang dilabelkan sebagai teks berbahasa Indonesia mengandung beberapa kata bahasa Inggris).

4.5.1. Tanpa feature selection

```
=== Run information ===

Scheme:         weka.classifiers.misc.InputMappedClassifier -I -trim -W
weka.classifiers.bayes.BayesNet -- -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S
BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

Relation:       Feature-Boolean-1-gram-weka.filters.unsupervised.attribute.Remove-R1

Instances:      8117

Attributes:   493

                [list of attributes omitted]

Test mode:      user supplied test set:  size unknown (reading incrementally)

Time taken to test model on supplied test set: 0.14 seconds



=== Summary ===

Correctly Classified Instances         192             74.7082 %

Incorrectly Classified Instances       65              25.2918 %

Kappa statistic                        0.6657

Mean absolute error                    0.1336

Root mean squared error                0.2681

Relative absolute error                36.3003 %

Root relative squared error            60.8299 %

Total Number of Instances              257


=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area  Class

                0.883    0.033    0.919    0.883    0.901    0.860    0.971    0.931    Inggris
```

```
                0.582   0.000   1.000   0.582   0.736   0.701   0.947   0.849   Indonesia
                0.642   0.000   1.000   0.642   0.782   0.766   0.943   0.808   Perancis
                0.917   0.282   0.427   0.917   0.583   0.504   0.924   0.771   Spanyol
Weighted Avg.   0.747   0.063   0.869   0.747   0.766   0.725   0.949   0.851


=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 68  0  0  9 |  a = Inggris
  1 46  0 32 |  b = Indonesia
  1  0 34 18 |  c = Perancis
  4  0  0 44 |  d = Spanyol
```

4.5.2. Dengan feature selection

```
=== Run information ===

Scheme:         weka.classifiers.misc.InputMappedClassifier -I -trim -W
weka.classifiers.bayes.BayesNet -- -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S
BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

Relation:
Feature-Boolean-1-gram-weka.filters.unsupervised.attribute.Remove-R1-weka.filters.supervised.attr
ibute.AttributeSelection-Eweka.attributeSelection.InfoGainAttributeEval-Sweka.attributeSelection.
Ranker -T 0.0 -N 246

Instances:      8117

Attributes:   247

                [list of attributes omitted]

Test mode:      user supplied test set:  size unknown (reading incrementally)

Time taken to build model: 0.73 seconds


=== Evaluation on test set ===
```

```
Time taken to test model on supplied test set: 0.09 seconds


=== Summary ===

Correctly Classified Instances         181               70.428  %
Incorrectly Classified Instances        76               29.572  %
Kappa statistic                          0.5906
Mean absolute error                      0.1635
Root mean squared error                  0.3235
Relative absolute error                 44.4111 %
Root relative squared error             73.3838 %
Total Number of Instances              257


=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area  Class
               1.000    0.422    0.503    1.000    0.670    0.539    0.894    0.755    Inggris
               0.570    0.000    1.000    0.570    0.726    0.692    0.899    0.781    Indonesia
               0.642    0.000    1.000    0.642    0.782    0.766    0.917    0.771    Perancis
               0.521    0.000    1.000    0.521    0.685    0.685    0.897    0.688    Spanyol
Weighted Avg.  0.704    0.127    0.851    0.704    0.713    0.660    0.901    0.754


=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 77  0  0  0 |  a = Inggris
 34 45  0  0 |  b = Indonesia
 19  0 34  0 |  c = Perancis
```

```
 23  0  0 25 |  d = Spanyol
```

4.5.3. Kesimpulan untuk eksperimen 5

Table 4. Perbandingan performa antara ekstraksi fitur tanpa *feature selection* dan dengan *feature selection*

Feature selection	*Accuracy (TP Rate)*	*Precision*	*Recall*	*F-Measure*	*ROC Area*	*Waktu (dalam sekon)*
Tanpa *feature selection*	0.747	0.869	0.747	0.766	0.949	1.84
Dengan *feature selection*	0.704	0.851	0.704	0.713	0.901	0.73

Berdasarkan eksperimen 5, didapatkan bahwa secara keseluruhan, klasifikasi dengan menggunakan *classifier Bayesian network* dengan ekstraksi fitur boolean 1-gram tanpa menggunakan *feature selection* memiliki performa yang sedikit lebih baik dibandingkan dengan menggunakan *feature selection* dengan metode *information gain,* meskipun waktu yang dibutuhkan untuk pembuatan model dengan menggunakan *feature selection* lebih sedikit dibandingkan dengan tidak menggunakan *feature selection.*

Penurunan akurasi dari eksperimen 4 (uji *training*) dan eksperimen 5 (uji *testing*) dapat disebabkan oleh ukuran *dataset* yang digunakan untuk *training* yang tidak besar, sehingga ketika diuji dengan *testing set* yang baru dan belum pernah diobservasi oleh model sebelumnya, maka akurasi akan menurun karena terdapat banyak fitur pada *training set* yang belum pernah dipelajari model sebelumnya.

5. Kesimpulan

Pada laporan ini, penulis telah melakukan eksperimen mengenai identifikasi bahasa secara otomatis dengan menggunakan klasifikasi. Eksperimen telah dilakukan pada tiga jenis *classifier* yang berbeda, yaitu *decision tree, neural network,* dan *Bayesian network.* Selain itu, dilakukan juga percobaan terhadap berbagai metode untuk melakukan ekstraksi fitur, yaitu ekstraksi fitur dengan jenis boolean dan frekuensi serta variasi n-gram dengan nilai n = 1, 2, 3, dan 4. Selanjutnya, dilakukan eksperimen mengenai penggunaan *feature selection* dengan metode *information gain.*

Berdasarkan kelima eksperimen yang telah dilakukan, didapatkan bahwa *Bayesian network* merupakan jenis *classifier* terbaik di antara ketiga *classifier* yang telah digunakan, dengan akurasi mencapai 92.449% dan nilai *f-measure* mencapai 0.908 dengan uji validasi menggunakan *10-fold cross validation*.

Selain itu, ditemukan bahwa jenis ekstraksi fitur dengan menggunakan boolean dan frekuensi pada *classifier Bayesian network* menghasilkan nilai akurasi, *precision, recall, F-measure,* dan *ROC Area* yang sama. Selain itu, nilai n untuk n-gram yang menghasilkan performa maksimal adalah n = 1. Hal ini dapat disebabkan oleh *dataset* yang terlalu kecil sehingga terdapat banyak kombinasi kata pada n-gram dengan n = 2, 3, dan 4 yang muncul pada *testing set* namun tidak muncul pada *training set.* Sehingga, untuk selanjutnya, disarankan untuk menggunakan *dataset* yang lebih besar agar proses klasifikasi dapat memanfaatkan ekstraksi

fitur n-gram untuk membuat model yang fitur-fiturnya berdasarkan dari kombinasi antar kata.

Selanjutnya, didapatkan bahwa penggunaan *feature selection* dengan metode *information gain* tidak memberikan hasil yang lebih baik dibandingkan dengan klasifikasi tanpa menggunakan *feature selection,* meskipun hasil dari kedua eksperimen tidak jauh berbeda dan penggunaan *feature selection* dapat mempercepat waktu pembuatan model. Hal ini menunjukkan bahwa meskipun dalam eksperimen ini penggunaan *feature selection* tidak meningkatkan nilai akurasi maupun *f-measure, feature selection* dengan metode *information gain* dapat dimanfaatkan untuk proses klasifikasi dengan *dataset* yang besar, karena dapat mempersingkat waktu pembuatan model dengan hasil nilai akurasi dan *f-measure* yang tidak jauh berbeda dengan *feature selection.* Selain itu, untuk selanjutnya, dapat juga digunakan metode *feature selection* lain yang dapat memberikan hasil lebih baik dari *information gain.*

Kesimpulan lain yang didapatkan pada eksperimen adalah terdapat penurunan akurasi antara eksperimen dengan uji coba menggunakan *training set* dengan *testing set.* Salah satu penyebab yang mungkin adalah ukuran *dataset* yang kurang besar, sehingga terdapat banyak fitur pada *testing set* yang belum pernah dipelajari oleh model sebelumnya.

References

Claus, David. *Handwritten Digit Recognition: Neural Network Classifier*. Information Engineering University of Oxford, n.d., http://www.robots.ox.ac.uk/~dclaus/digits/neural.htm. Diakses 5 Desember 2016.

Domingos, Pedro. *A Few Useful Things to Know about Machine Learning*. Communications of the ACM, 55 (10), 78-87, 2012, https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf. Diakses 5 Desember 2016.

Ethem, A., 2010. *Introduction to Machine Learning*. MIT Press, p. 9.

"*Evaluate model performance in Machine Learning". Microsoft Azure,* 19 Agustus 2016, www.docs.microsoft.com/en-us/azure/machine-learning/machine-learning-evaluate-model-performance#evaluating-a-binary-classification-mode. Diakses 5 Desember 2016.

T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

Pearl, J., 1998. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Stiller, J., Gäde, M., Petras, V., 2010. *Ambiguity Of Queries And The Challenges For Query Language Detection*. CLEF 2010 Working Notes, Berlin, 2010, http://ceur-ws.org/Vol-1176/CLEF2010wn-LogCLEF-StillerEt2010.pdf. Diakses 5 Desember 2016.

Schneider, Jeff. *Cross Validation*. Carnegie Mellon School of Computer Science, n.d., www.cs.cmu.edu/~schneide/tut5/node42.html. Diakses 5 Desember 2016.

Schapire, Rob. *Machine Learning Algorithms for Classification*. Princeton University, n.d., http://www.cs.princeton.edu/~schapire/talks/picasso-minicourse.pdf. Diakses 5 Desember 2016.

Sulhan. Pembelajaran Pohon Keputusan *(Decision Tree)*. STEI ITB, www.sites.google.com/a/std.stei.itb.ac.id/sulhan/pembelajaran-mesin-lanjut/pembelajaran-pohon-keputusan-decision-tree. Diakses 5 Desember 2016.