Question Classification Menggunakan Support Vector Machines dan Stemming

Abdiansah
Laboratorium Sistem Cerdas, UGM
Departemen Ilmu Komputer, Universitas Sriwijaya
Email: abdiansah@unsri.ac.id

Edi Winarko
Laboratorium Sistem Cerdas, UGM
Departemen Ilmu Komputer dan Elektronika, Universitas Gadjah Mada
Email: ewinarko@ugm.ac.id

***Abstract*—*Question Classification* (QC) merupakan salah satu komponen penting dalam *Question Answering System* (QAS) karena akan berpengaruh langsung terhadap kinerja keseluruhan QAS. Sejauh ini metode yang disarankan oleh komunitas QAS untuk QC adalah menggunakan *Support Vector Machines* (SVM). Untuk melakukan klasifikasi teks dibutuhkan fitur berdimensi tinggi, banyaknya fitur dapat mengurangi performa SVM. *Stemming* adalah teknik yang digunakan untuk mereduksi term suatu dokumen. Penggunaan *stemming* akan berpengaruh terhadap sintaksis dan semantik suatu pertanyaan. Penelitian ini bertujuan untuk mengetahui pengaruh *stemming* terhadap akurasi SVM. Telah dilakukan dua percobaan klasifikasi pertanyaan, yaitu dengan menggunakan SVM dan SVM+*stemming*. Hasil rata-rata akurasi dari percobaan diperoleh sebesar 86.75% untuk SVM dan 87.48% SVM+*stemming* sehingga telah terjadi kenaikan akurasi sebesar 0.73%. Walaupun peningkatan akurasi tidak signifikan tetapi *stemming* dapat mereduksi fitur tanpa menurunkan akurasi SVM.**

Keywords*—question classification, question answering system, support vector machines, *stemming

I. PENDAHULUAN

Umumnya proses pencarian informasi di internet menggunakan mesin pencari. Pengguna memberikan kata kunci kemudian mesin pencari akan melakukan proses *crawling* yaitu pencarian tautan-tautan situs yang ada dalam basis data. Hasil dari pencarian berupa daftar situs-situs relevan dengan kata kunci pengguna. Sayangnya, mesin pencari hanya memberikan rujukan situs-situs yang relevan, selebihnya pengguna disuruh mencari sendiri informasi yang dibutuhkan. *Question Answering System* (QAS) merupakan sistem yang menerima pertanyaan pengguna dalam bahasa alami dan memberikan jawaban yang tepat, misalnya terdapat pertanyaan "*siapa nama Presiden Indonesia?*" maka sistem akan memberikan jawaban: "*Soekarno*". Oleh sebab itu QAS merupakan sistem yang tidak hanya mencari informasi seperti mesin pencari, tetapi juga dapat memberikan jawaban langsung kepada pengguna.

Terdapat tiga komponen penyusun QAS yaitu *question analysis*, *information retrieval* dan *answer extraction*. Salah satu sub komponen yang ada dalam *question analysis* adalah *question classification* yang berfungsi untuk memprediksi *Expected Answer Type* (EAT) suatu pertanyaan. EAT merupakan kategorisasi yang digunakan untuk menentukan kelas suatu pertanyaan, misalnya pada pertanyaan contoh sebelumnya yang arah jawabannya menanyakan nama orang maka EAT dari pertanyaan tersebut adalah PERSON. Dengan diketahuinya EAT maka proses pencarian jawaban dapat direduksi. Salah satu metode *question classification* yang disarankan oleh komunitas QAS adalah *Support Vector Machines* (SVM).

Question Classification (QC) merupakan bagian penting dari QAS karena akan berpengaruh langsung terhadap akurasi dari ektraksi jawaban serta menentukan kualitas dan unjuk kerja keseluruhan dari QAS [1]. QC berfungsi untuk memahami arah pertanyaan yang diajukan, misalnya ada pertanyaan: "*Who was the first American in space?*" dari pertanyaan tersebut dapat dipastikan bahwa yang ditanyakan adalah nama orang. *Stemming* adalah proses untuk menjadikan suatu kata yang sudah berubah bentuk menjadi kata dasar. S*temming* biasanya digunakan sebagai pra-pengolahan untuk *Information Retrieval* (IR) yang berfungsi untuk mereduksi jumlah term dalam suatu dokumen. Misalnya terdapat empat term, "*running*", "*runs*", "*runned*", "*runly*", setelah dilakukan proses *stemming* maka keempat term tersebut menjadi satu term yaitu "*run*" karena kata dasar dari keempat term tersebut adalah "*run*".

Artikel ini berisi laporan penelitian tentang pengaruh penggunaan *stemming* sebagai pra-pengolahan untuk *question classification* menggunakan metode SVM. Efek dari penggunaan *stemming* akan berpengaruh terhadap sintaksis dan semantik suatu kalimat. Selain itu, *stemming* dapat mengurangi dimensi fitur, yang dapat menyebabkan menurunnya performa SVM. Oleh karena itu tujuan penelitian ini adalah untuk mengetahui pengaruh *stemming* terhadap akurasi SVM. Jika akurasi tetap maka sistem akan mendapatkan keunggulan dari sisi reduksi dimensi fitur karena *stemming* berfungsi untuk mereduksi term yang ada dalam suatu corpus. Sedangkan jika akurasi bertambah maka sistem akan mendapatkan dua keunggulan yaitu reduksi dimensi fitur dan akurasi. Artikel ini disusun sebagai berikut, Bagian 2 berisi penelitian terdahulu. Bagian 3 menjelaskan metode dan implementasinya. Bagian 4 menjelaskan skenario percobaan yang dilakukan, hasil dari percobaan tersebut akan dijelaskan di Bagian 5 dan kesimpulan ada pada Bagian 6.

II. PENELITIAN TERKAIT

Question Classification berfungsi untuk memahami arah pertanyaan yang diajukan, misalnya ada pertanyaan: "*Who was the first American in space?*" dari pertanyaan tersebut dapat dipastikan bahwa yang ditanyakan adalah nama orang. QC berfungsi untuk menentukan *Expected Answer Type* (EAT) sehingga dapat mereduksi ruang pencarian [6]. EAT

merupakan label yang diberikan untuk suatu pertanyaan, misalnya dari contoh sebelumnya maka EAT-nya adalah HUMAN [10].

Secara umum terdapat tiga pendekatan metode yang digunakan untuk QC yaitu: 1) pendekatan *Knowledge-Based* (KB); 2) pendekatan *Machine Learning* (ML) dan 3) *Hybrid*. Beberapa penelitian yang menggunakan pendekatan KB diantaranya dilakukan oleh [1] menggunakan *rule-based* dan analisis sintaksis yang fokus pada pertanyaan *5WH* (*why, when, where, which, who* dan *how*). Hasil akurasi rata-rata mencapai 97.5%. [2] menggunakan *regex* dan *sememes model*, dengan fitur *interrogative words*, *question focus word*, dan *first sememes*. Artikel menyebutkan bahwa hasil percobaan menunjukkan lebih baik dari *pattern matching* dan *machine learning approach*. [16] menggunakan *Association Rule* (AR) dengan fitur sederhana: *words* dan *bi-gram*. Hasil penelitian menunjukan bahwa AR dapat digunakan untuk QC dan memberikan performa yang bagus sekitar 86.4%. [14] dan [15] melakukan penelitian menggunakan *Semantic Approach* (SA) dan *Semantic Pattern* (SP). SA menggunakan WordNet dan term yang ada di Wikipedia. Kombinasi antara WordNet dan Wikipedia dapat menghasilkan akurasi QC sebesar 89.5%.

Penelitian QC menggunakan ML dilakukan oleh [10][11] menggunakan model SNoW dan mengenalkan konsep hirarki klasifikasi. Mereka mengusulan dua kelas klasifikasi yaitu: 6 *coarse-class* dan 50 *fine-class*. Hirarki pertanyaan dengan dua layer yang dibuat oleh [10] dijadikan sebagai standar untuk penelitian QC. Dataset yang digunakan diambil dari UIUC sebanyak 5.500 data untuk *training* dan dari TREC sebanyak 500 data untuk *testing*. Dari hasil pengujian didapat akurasi sebesar 98.0% untuk *coarse-class* dan 95.0% untuk *fine-class*. [8] dan [20] menggunakan SVM sebagai metode klasifikasi dengan kernel linear dimana akurasi yang diperoleh oleh [8] sebesar 89.2% untuk *fine-class* dan 93.4% untuk *coarse-class*. Sedangkan [20] memperoleh akurasi sebesar 90.8% untuk *fine-class* dan 95.0% untuk *coarse-class*.

Metode ML dapat di-*hybrid* dengan metode lain guna mencari hasil yang optimal. Ada beberapa penelitian yang telah melakukannya diantaranya dilakukan oleh [3] yang menggunakan metode INFOMAP. *Hybrid Knowledge-based Approach* (INFOMAP) dengan SVM dapat meningkatkan akurasi klasifikasi sampai 92.0%. [12] menggunakan algoritma semi-supervisi *Tri-Training* dan SVM. [4] menggunakan *syntactic Feature* dan *semantic Feature* dengan metode *hybrid*: GA (*Genetic Algorithm*) untuk *feature selection*, CRF (*Conditional Random Field*) untuk *question informer prediction* dan SVM untuk *question classification*. *Hybrid* dari arsitektur (GA-CRF-SVM) dapat meningkatkan QA dengan memprediksi *question informer*. [21] menggunakan *Latent Semantic Analysis* (LSA) untuk mereduksi fitur sehingga ukurannya menjadi lebih kecil dan efisien. Mereka menggunakan *Backpropagation Neural Network* (BPNN) dan SVM untuk mengevaluasi dan didapatkan akurasi BPNN lebih unggul dibanding SVM untuk ukuran fitur yang sedikit

Dari hasil pengamatan penelitian sebelumnya dapat dilihat bahwa metode SVM memberikan hasil akurasi yang cukup bagus, tetapi SVM dapat mengalami penurunan performa ketika menggunakan fitur berdimensi tinggi. Untuk mengatasi masalah tersebut dapat digunakan teknik seleksi fitur. *Stemming* dapat juga dilihat sebagai teknik seleksi fitur yang sederhana. Oleh karena itu, tujuan penelitian ini mencoba menerapkan *stemming* sebagai pra-pengolahan untuk SVM dan melihat akurasi SVM-*baseline* dan SVM-*stemming*.

III. METODOLOGI

Bagian ini akan menjelaskan metodologi yang digunakan dalam penelitian ini dimulai dari Analisis Data, *Stemming*, *Bag-of-Words*, *Support Vector Machines* dan Kakas Implementasi.

A. Analisis Data

Dalam penelitian ini, sumber data[1] diambil dari penelitian [10] dengan data latih dibuat 4 corpus yang masing-masing corpus terdiri dari 1.000, 2.000, 3.000, 4.000 pertanyaan dan data uji sebanyak 1.000 pertanyaan. Semua data tersebut sudah ada label yang dibuat oleh oleh [10] dan menggunakan dua leyer. Data uji adalah subset dari data latih, bahasa yang digunakan adalah bahasa inggris. Setiap term dari suatu corpus akan dijadikan atribut klasifikasi, sedangkan untuk kelas klasifikasi digunakan layer satu atau 6 kelas (*coarse class*) yang merujuk pada penelitian [10] yaitu: *Abbreviation, Description, Entity, Human, Location* dan *Numeric*.

B. Stemming

Stemming sering digunakan dalam sistem *Information Retrieval* (IR) untuk mengubah suatu kata menjadi akar kata [19]. *Stemming* berfungsi untuk mereduksi jumlah term suatu dokumen. Algoritma yang digunakan adalah algoritma Potter yang diambil dari pustaka snowball[2]. Berdasarkan penelitian yang dilakukan oleh [9], hasil *stemming* dari snowbal secara keseluruhan cukup bagus.

C. Bag of Words Model

Model *Bag of Words* (BoW) merupakan model representasi teks paling sederhana. Model ini akan mengabaikan urutan kata [17]. Penggunaan BoW dalam penelitian ini hanya untuk kemudahan representasi teks sehingga terlepas dari pengaruh sintaksis dan semantik. Suatu pertanyaan tersusun dari term-term, sehingga setiap *term* berisi jumlah term yang ada dalam dokumen (*term frequency*). Term tersebut akan dijadikan fitur-fitur klasifikasi yang sehingga banyaknya fitur dalam suatu *corpus* ditentukan oleh banyaknya term yang ada dalam *corpus* tersebut.

D. Support Vector Machines

Support Vector Machines (SVM) merupakan salah satu algoritma *machine learning* yang menggunakan model *supervised learning* untuk mengenali suatu pola. SVM sering digunakan untuk klasifikasi dan analisis regresi. Berdasarkan penelitian dari Zheng & Lee (2003) menunjukan bahwa penggunaan SVM untuk *question classification* cukup bagus dengan akurasi di atas 80.0%.

Misalnya diberikan sebuah *training set,*

$$(X_i, Y_i), i = 1, \ldots, n,$$

dimana,

$$X_i = (x_i, \ldots, x_{id})$$

1 http://cogcomp.cs.illinois.edu/Data/QA/QC
2 http://snowball.tartarus.org

merupakan sebuah sampel dimensi d dan $y_i \in \{1,-1\}$ adalah label yang diberikan terhadap sampel. Tugas SVM menemukan fungsi diskriminan linear $g(x)=w^T X+w_0$ sehingga,

$$w^T x_i + w_0 \geq +1 \quad \text{untuk} \quad y_i = +1$$
$$w^T x_i + w_0 \leq -1 \quad \text{untuk} \quad y_i = -1$$

Solusi dari permasalahan tersebut harus memenuhi persamaan berikut:

$$y_i(w^T x_i + w_0) \geq 1 \qquad i=1,\ldots,n \qquad (1)$$

Fungsi linear optimal dapat diperoleh dengan meminimalkan masalah pemrograman kuadratik berikut ini [13]:

$$min \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha(y_1(w^T x_i + w_0) - 1) \qquad (2)$$

yang akan menghasilkan solusi berikut ini:

$$w = \sum_{i=1}^{n} \alpha y_i x_i \qquad (3)$$

dimana, $\{\alpha, i=1,\ldots,n; \alpha \geq 0\}$ adalah Lagrange multipliers.

Untuk memungkinkan data terpisah secara linear, biasanya ruang fitur dipetakan ke dalam ruang berdimensi tinggi. Teknik yang digunakan untuk melakukan pemetaan tersebut disebut dengan fungsi Kernel. Kernel adalah sebuah fungsi,

$$k: \chi \times \chi \rightarrow \mathbb{R}$$

yang mengambil dua sampel dari ruang masukan dan memetakan menjadi bilangan real yang mengindikasikan tingkat kesamaannya. Untuk semua,

$$x_i, x_j \in \chi,$$

maka fungsi Kernel harus memenuhi:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \qquad (4)$$

dimana ϕ merupakan pemetaan eksplisit dari ruang masukan χ menjadi fitur *dot product* ruang H [5].

Untuk menerapkan fungsi Kernel pada SVM, umumnya persamaan (2) diselesaikan dengan persamaan berikut:

$$max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \qquad (5)$$

dimana $x_i \cdot x_j$ adalah inner product dari dua sample yang merupakan kernel implisit dalam persamaan ukuran kemiripan antara x_i dan x_j. Inner product tersebut dapat diganti dengan fungsi Kernel lain sehingga persamaan (5) akan menjadi seperti persamaan berikut:

$$max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \qquad (6)$$

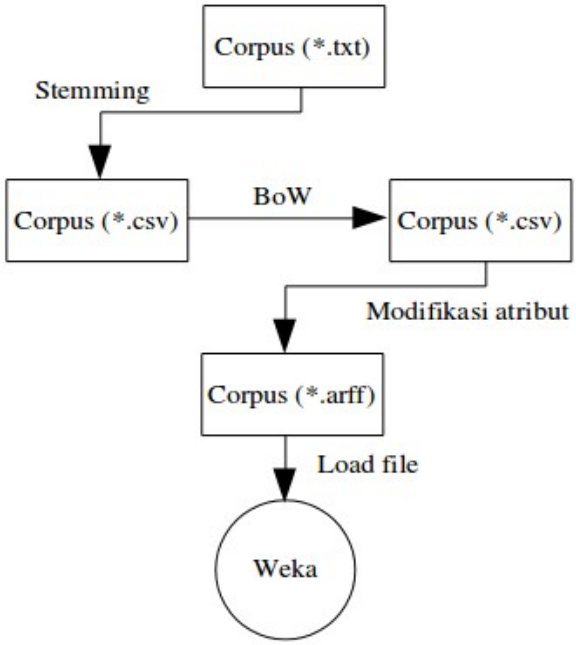
Terdapat empat tipe Kernel dasar: *linear*, *polynomial*, *radial basis function* dan *sigmoid*. SVM dapat digunakan untuk multi-class, yaitu menggunakan strategi *one-against-one* [18] yang sudah diuji oleh [7] dan hasilnya cukup baik.

E. Kakas Implementasi

Ada dua kakas yang digunakan dalam penelitian ini yaitu: 1) *snowball stemming*, yang berbentuk library dalam bahasa java (*.jar) dan 2) *Weka 3.71* yang digunakan untuk melakukan percobaan pelatihan dan pengujian SVM. Untuk proses *stemming* dibuat aplikasi berbasis java yang memanggil pustaka snowball. Masukan dan luaran dari aplikasi tersebut berbentuk file teks. Proses pelatihan dan pengujian SVM menggunakan LibSVM yang tersedia di Weka. Modifikasi parameter hanya untuk variabel gamma yang sebelumnya bernilai 0.0 diganti menjadi 0.5. Penggunaan parameter tersebut berdasarkan uji coba *trial and error*. Kakas ini digunakan di komputer Intel Celeron B815 - 1.60 Ghz (2 CPU), RAM – 3.8 GiB, sedangkan sistem operasi menggunakan LINUX distro Kubuntu 14.04 LTS – 32 bit.

IV. SKENARIO UJI COBA

Sebelum data diproses menggunakan Weka, dilakukan dulu proses konversi data yang dimulai dari file teks (*.txt) yang berisi data pertanyaan menjadi file csv (*.csv) yang berisi data pertanyaan hasil *stemming* serta sudah direpresentasikan dalam bentuk model BoW. Berikutnya file csv tersebut di konversi menjadi file arff (*.arff) supaya bisa digunakan Weka. Pada dasarnya Weka bisa mengenali file csv otomatis hanya saja untuk pengujian diperlukan kesamaan atribut antara data latih dan data uji, sehingga harus ada modifikasi atribut yang hanya bisa dilakukan dalam format arff.



Gambar 1. Tahapan konversi data corpus

Pada Gambar 1 dapat dilihat tahapan pemrosesan data sebelum dilakukan pelatihan. Percobaan dilakukan dua kali yaitu tanpa *stemming* dan menggunakan *stemming*. Untuk percobaan tanpa *stemming*, proses konversi data langsung dari file teks ke file arff. Untuk tiap percobaan, masing-masing *corpus* akan dilatih dan diuji, sehingga terdapat 4 kali pelatihan dan pengujian. Data uji yang digunakan sama untuk masing-masing *corpus*. Pada Tabel 1 dapat dilihat representasi teks dalam bentuk BoW yang berisi nilai tiap term.

TABEL 1. REPRESENTASI TEKS MENGGUNAKAN BOW

Pertanyaan	**what**	**is**	**was**	**the**	**...**
what access is moxie	1	1	0	0	...
what is her profession	1	1	0	0	...
what was about known as the spice island	1	0	1	0	...
...	...	...	...	...	...

TABEL 2. JUMLAH ATRIBUT MASING-MASING CORPUS

Percobaan SVM	**Corpus**			
	1000	**2000**	**3000**	**4000**
Tanpa Stemming	2.481	4.075	5.061	6.189
Stemming	2.210	3.469	4.242	5.115
Reduksi (%)	10.92	14.87	16.18	17.35

Pada Tabel 2 dapat dilihat total *term* untuk masing-masing *corpus* dengan dua percobaan, tanpa *stemming* dan menggunakan *stemming*. Berdasarkan percobaan yang dilakukan didapat rata-rata reduksi untuk seluruh *corpus* sebesar 14.83% dan semakin banyak data *corpus* maka persentasi reduksi akan semakin besar. Dari hasil pengamatan dapat disimpulkan bahwa hasil reduksi dipengaruhi oleh *term* dan *corpus*. Jumlah *corpus* yang besar dengan *term* yang sedikit akan menaikan persentase reduksi, sebaliknya jumlah *corpus* yang sedikit dengan term yang banyak akan menurunkan persentase reduksi.

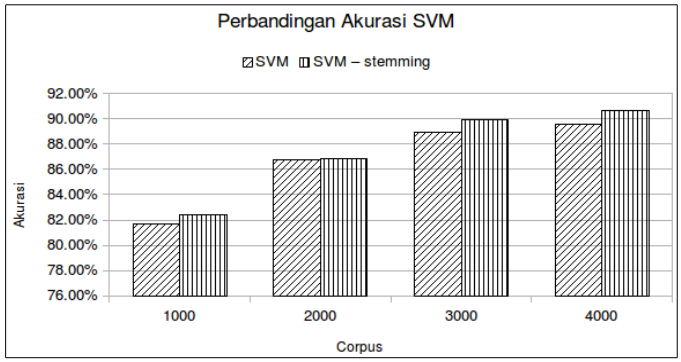
Pada Tabel 3 dapat dilihat hasil percobaan terhadap akurasi SVM tanpa dan menggunakan *stemming*. Pada tabel tersebut dapat dilihat terjadi peningkatan akurasi walaupun tidak terlalu signifikan yaitu sebesar 0.73% untuk rata-rata akurasi semua *corpus*. Walaupun kenaikan akurasi kecil tapi penggunaan *stemming* dapat mereduksi atribut sehingga membantu kecepatan proses pelatihan dan pengujian.

TABEL 3. AKURASI SVM UNTUK TIAP CORPUS

Percobaan SVM	**Corpus**			
	1000	**2000**	**3000**	**4000**
Tanpa Stemming (%)	81.70	86.80	88.90	89.60
Stemming (%)	82.40	86.90	89.90	90.70
Selisih (%)	0.70	0.10	1.00	1.10

V. HASIL UJI COBA

Berdasarkan percobaan maka dapat diperoleh hasil akurasi untuk SVM tanpa *stemming* dan SVM menggunakan *stemming*. Pada gambar 2 dapat dilihat perbandingan hasil akurasi SVM, tampak pada Gambar 2 bahwa terjadi kenaikan akurasi setelah menggunakan *stemming*. Dari hasil percobaan didapat juga total akurasi untuk masing-masing *corpus*, dimana akurasi SVM tanpa *stemming* (*baseline*) sebesar 86.75% sedangkan SVM *stemming* sebesar 87.48% sehingga telah terjadi kenaikan akurasi sebesar 0.73%. Hasil tersebut juga menunjukan bahwa SVM memberikan hasil yang cukup baik untuk QC.



Gambar 2. Perbandingan akurasi SVM

VI. KESIMPULAN

Question Classification (QC) merupakan tahapan awal dalam *Question Answering System* (QAS) yang memberikan pengaruh besar dalam menentukan keberhasilan sistem untuk mencari jawaban yang benar. Oleh karena itu penelitian di bidang QC terus dieksplorasi guna mendapatkan hasil yang maksimal. Komunitas QAS mengusulkan menggunakan *Support Vector Machines* (SVM) untuk menentukan kelas suatu pertanyaan. Dari hasil percobaan yang dilakukan, SVM memberikan hasil yang cukup bagus, terlebih lagi ditambah tahapan *stemming* yang dapat mereduksi data tanpa menurunkan tingkat akurasi SVM. Oleh karena dapat disimpulkan bahwa tahapan *stemming* dapat digunakan SVM untuk menentukan kelas suatu pertanyaan.

REFERENSI

[1] Biswas, P., Sharan, A., & Kumar, R. (2014). Question Classification using syntactic and rule based approach. In *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on* (pp. 1033-1038). IEEE.

[2] Cai, D., Bai, Y., Dong, Y., & Liu, L. (2007). Chinese Question Classification Using Combination Approach. In Semantics, Knowledge and Grid, Third International Conference on (pp. 334-337). IEEE.

[3] Day, M. Y., Lee, C. W., Wu, S. H., Ong, C. S., & Hsu, W. L. (2005). An integrated knowledge-based and machine learning approach for Chinese question classification. In Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on (pp. 620-625). IEEE.

[4] Day, M. Y., Ong, C. S., & Hsu, W. L. (2007). Question classification in english-chinese cross-language question answering: an integrated genetic algorithm and machine learning approach. In Information Reuse and Integration, 2007. IRI 2007. IEEE International Conference on (pp. 203-208). IEEE.

[5] Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The annals of statistics*, 1171-1220.

[6] Hovy, E., Gerber, L., Hermjakob, U., Lin, C. Y., & Ravichandran, D.

(2001). Toward semantics-based answer pinpointing. In Proceedings of the first international conference on Human language technology research (pp. 1-7). Association for Computational Linguistics.

[7] Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, *13*(2), 415-425.

[8] Huang, P., Bu, J., Chen, C., Qiu, G., & Zhang, L. (2007). Learning a Flexible Question Classifier. In Convergence Information Technology, 2007. International Conference on (pp. 1608-1613). IEEE.

[9] Kamps, J., Monz, C., de Rijke, M., & Sigurbjörnsson, B. (2003). Approaches to Robust and Web Retrieval. In *TREC* (pp. 594-599).

[10] Li, X., & Roth, D. (2002, August). Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1* (pp. 1-7). Association for Computational Linguistics.

[11] Li, X., & Roth, D. (2006). Learning question classifiers: the role of semantic information. Natural Language Engineering, 12(03), 229-249.

[12] Nguyen, T. T., Nguyen, L. M., & Shimazu, A. (2007). Improving the Accuracy of Question Classification with Machine Learning. In Research, Innovation and Vision for the Future, 2007 IEEE International Conference on (pp. 234-241). IEEE.

[13] Vapnik, V. (2000). *The nature of statistical learning theory*. Springer Science & Business Media.

[14] Ray, S. K., Singh, S., & Joshi, B. P. (2010). A semantic approach for question classification using WordNet and Wikipedia. Pattern Recognition Letters, 31(13), 1935-1943.

[15] Song, W., Wenyin, L., Gu, N., Quan, X., & Hao, T. (2011). Automatic categorization of questions for user-interactive question answering. Information Processing & Management, 47(2), 147-156.

[16] Sun, Y. L. (2010). Chinese question classification based on mining association rules. In ICMLC (pp. 413-416).

[17] Wallach, H. M. (2006, June). Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning* (pp. 977-984). ACM.

[18] Webb, A. R. (2002). Statistical pattern recognition, 2nd Edition. John Wiley & Sons.

[19] Xu, J., & Croft, W. B. (1998). Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems (TOIS)*, *16*(1), 61-81.

[20] Silva, J. Coheur, L. Mendes, A and Wichert, A. (2011). From symbolic to sub-symbolic information in question classification. Artificial Intelligence Review, 35(2):137–154, February.

[21] Loni, B., Khoshnevis, S. H., & Wiggers, P. (2011). Latent semantic analysis for question classification with neural networks. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on* (pp. 437-442). IEEE.