Entity-based Oversampling to Handle Class Imbalance Problem in Named Entity Recognition

Swardiantara Silalahi

Abstrak

Named Entity Recognition merupakan salah satu permasalahan klasifikasi dengan data yang digunakan berupa text. Domain ini melakukan prediksi terhadap kata-kata pada kalimat yang memiliki arti dalam konteks tertentu. Dengan melakukan pengenalan terhadap kata-kata penting tersebut, dapat dilakukan ekstraksi informasi yang terkandung dalam kalimat. Topik penelitian klasifikasi memiliki salah satu permasalahan yang sering ditemui oleh peneliti, yaitu *Class Imbalance Problem* (CIP). Kondisi di mana komposisi data antar kelas yang sangat timpang ini berpotensi menghasilkan model klasifikasi yang kurang optimal. Pada domain NER, kata-kata yang dianggap tidak penting akan dilabeli dengan tag "O" yang berarti *Outside*. Populasi kata dengan tag "O" pada kalimat jauh lebih besar dibandingkan kata yang merupakan entitas penting. Pada penelitian ini, dilakukan percobaan untuk mengembangkan satu teknik *oversampling* untuk mengatasi CIP pada domain NER. *Entity-based oversampling* dilakukan untuk menambah populasi data yang merupakan entitas dengan tujuan agar model dapat lebih baik mengenali entitas-entitas ini dalam kalimat. Untuk melakukan klasifikasi, BiLSTM dan BiGRU digunakan dengan ukuran padding 50 dan embedding 100. Metode oversampling yang dikembangkan belum mampu meningkatkan performa model. Diperlukan penelitian lebih lanjut untuk memperbaiki model agar dapat meningkatkan performa model klasifikasi yang dibangun.

PENDAHULUAN

Teknik resampling pada domain NER memiliki karakteristik berbeda dibandingkan dengan resampling atau augmentasi pada data tabular atau gambar. Data sumber pada NER berupa kumpulan kalimat yang berisi kata-kata dengan tag entitas tertentu untuk diprediksi. Entitas merupakan kata yang memiliki nilai pada domain tertentu, sehingga perlu dikenali secara otomatis untuk mengekstrak informasi yang terkandung pada sebuah teks. Entitas akan diberi tag B (*beginning*) dan I (*inside*), selain itu akan diberi tag O (*outside*) jika menggunakan format anotasi IOB. Kata dengan tag O akan disebut sebagai kelas negatif, sebaliknya kata dengan tag selain O akan disebut sebagai kelas positif. Salah satu teknik yang pernah dikembangkan adalah *stopwords removal* [1], yaitu dengan mengeliminasi kata-kata yang merupakan sering muncul pada kalimat dan dianggap kurang penting, dimana dapat mengurangi populasi data mayor dengan kelas negatif. Teknik ini diaplikasikan terhadap data *train* dan data *test*. Akkasi, Varoglu dan Dimililer [2] mengembangkan satu metode Balance Undersampling (BUS) untuk mengurangi data kelas negatif yang merupakan data dengan populasi mayor pada NER. Teknik ini menggunakan ratio antara kelas negatif dengan kelas positif sebagai acuan dalam mengeliminasi kata-kata yang merupakan anggota kelas "Outside". Kemudian menandai kata-kata dengan tag "Outside" yang berada di sekitar kata dengan tag selain "Outside" untuk dipertahankan dan menghapus kata lainnya. Luaran dari teknik ini akan menghasilkan dataset dengan komposisi data antara kelas positif dan negatif sesuai dengan ratio yang sudah ditentukan sebelumnya. Teknik ini terbukti mampu meningkatkan

performa model klasifikasi yang digunakan dibandingkan dengan teknik *Random Undersampling* (RUS) pada uji coba dengan menggunakan empat dataset yang berbeda. Untuk mendapatkan ratio yang optimal, Akkasi [3] menggunakan algoritma genetika pada saat melakukan proses undersampling. Fungsi fitness yang digunakan adalah skor evaluasi dari model klasifikasi yang digunakan. Teknik ini menghasilkan dataset dengan komposisi tertentu yang dapat mengoptimalkan performa dari model klasifikasi yang digunakan. Usaha dalam meningkatkan performa model pada domain NER tidak terbatas pada penanganan *class imbalance problem.* Salah satu usaha yang dapat dilakukan adalah dengan mengkombinasikan beberapa model klasifikasi untuk menghasilkan model yang dapat saling melengkapi. Akkasi [4] melakukan uji coba terhadap dataset ChemDNER untuk melihat efek dari *Random Undersampling* dengan bantuan *ensemble classifier* yang bekerja secara *majority voting.* Teknik ini menghasilkan model *ensemble* dengan performa yang mengungguli model-model *baseline* sebelum dikombinasikan. Penelitian yang sudah dilakukan untuk menangani permasalahan *class imbalance* pada NER masih didominasi oleh teknik *undersampling*. Pada penelitian ini, penulis mengusulkan sebuah metode dengan pendekatan *oversampling* dengan menambah populasi kata kelas positif untuk mengurangi ketimpangan komposisi pada dataset.

METODE

Teknik *entity-based oversampling* (EOS) ini melakukan pembangkitan kata dengan tag I yang akan dipasangkan ke setiap kata dengan tag B yang berdiri sendiri. Sehingga akan dihasilkan dataset dengan keberadaan entitas yang selalu berpasangan (Beginning-Inside). Pemilihan kandidat kata yang akan dibangkitkan diambil dari data train secara acak, dan proses pemilihan kata yang akan ditambahkan ke kata dengan tag B juga diambil secara acak. Dengan teknik ini, populasi kata dengan tag I sedikitnya akan sama dengan populasi kata dengan tag B. Proses ini secara sekilas dapat dilihat pada Gambar 1.

Sentence #	Word	Tag
Sentence: 1	Thousands	O
	of	O
	demonstrators	O
	have	O
	marched	O
	through	O
	London	B-geo
	to	O
	protest	O
	the	O
	war	O
	in	O
	Iraq	B-geo
	and	O
	demand	O
	the	O
	withdrawal	O

	Word	Tag
1	Word	Tag
2	Thousands	O
3	of	O
4	demonstrators	O
5	have	O
6	marched	O
7	through	O
8	London	B-geo
9	Province	I-geo
10	to	O
11	protest	O
12	the	O
13	war	O
14	in	O
15	Iraq	B-geo
16	Islands	I-geo
17	and	O
18	demand	O

Gambar 1. Entity-based Oversampling (EOS)

Oversampling hanya dilakukan terhadap data train saja, sehingga model yang dibangun akan belajar dari data hasil *oversampling* dengan data kalimat yang sudah berubah. Data ini memiliki populasi kelas positif yang lebih banyak dibandingkan data awal, dengan harapan model dapat mengenali kata-kata yang merupakan entitas. Model klasifikasi yang digunakan adalah BiLSTM dan BiGRU yang banyak digunakan pada penelitian Named Entity Recognition [5]. Arsitektur model klasifikasi yang digunakan dapat dilihat pada Gambar 2. Sebelum melakukan training model klasifikasi, data kalimat terlebih dahulu diubah menjadi data numerik dengan proses tokenisasi, *padding* dan *embedding*. Kata-kata pada setiap kalimat akan diubah menjadi vektor dengan panjang dimensi tertentu. Agar arsitektur *deep learning* dapat memproses kalimat, maka panjang kalimat juga perlu diseragamkan melalui proses *padding*. Dengan melihat distribusi panjang kalimat pada dataset, maka dipilih ukuran padding sebesar 50, agar data padding (biasanya berupa angka 0) dan kata yang dihilangkan seimbang. Dalam hal ini, data padding yang ditambahkan tidak terlalu banyak dibandingkan jika memilih ukuran padding dari panjang maksimal kalimat pada dataset.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 50, 100)	3517200
bidirectional_1 (Bidirectional)	(None, 50, 200)	160800
lstm_3 (LSTM)	(None, 50, 100)	120400
dropout_1 (Dropout)	(None, 50, 100)	0
dense_1 (Dense)	(None, 50, 17)	1717

Gambar 2. Arsitektur Model BiLSTM + LSTM

Pilihan *resampling* lain yang dapat dilakukan adalah dengan melakukan *resampling* terhadap data pada level representasi numerik (*embedding*). Beberapa metode yang dapat digunakan seperti SMOTE dan variansnya untuk oversampling, dan Tomek Links untuk undersampling. Metode ini akan membangkitkan data berupa vektor dengan representasi yang mirip dengan vektor asal. Proses ini dapat digunakan untuk menambah populasi data kelas minor (*oversampling*), ataupun untuk mengurangi populasi data kelas major (*undersampling*).

HASIL EKSPERIMEN

Uji coba dilakukan dengan menggunakan dataset NER yang diambil dari kaggle[1] dengan jumlah kalimat dan kata masing-masing adalah 47.959 dan 1.048.575. data ini kemudian dibagi menjadi data train dan data test dengan komposisi data test sebanyak 20% jumlah kalimat, yaitu 9.591 kalimat. Komposisi dataset untuk masing-masing tag dapat dilihat pada Tabel 1. Sebagai

[1] https://www.kaggle.com/namanj27/ner-dataset

perbandingan, pada penelitian ini dilakukan undersampling dengan pendekatan *stopwords filtering* (SWF) dan *punctuation filtering* (PF). Uji coba ini dilakukan pada Google Colab dengan lisensi standar. Keterbatasan ini membuat uji coba *resampling* pada level *embedding* tidak berhasil dilakukan.

Tabel 1. Perubahan komposisi data setelah *Undersampling* dan *Oversampling*

Tags	**Baseline**	**SWF**	**PF**	**EOS**
O	711029	453993	617884	711029
B-geo	30162	29976	**27063**	30162
I-geo	5959	**5803**	5833	**28961**
B-per	13667	13667	**10841**	13667
I-per	13917	13906	13334	**16843**
B-org	16125	16077	**14237**	16125
I-org	13583	**12631**	**12849**	**21439**
B-tim	16251	**15776**	**12571**	16251
I-tim	5269	**4493**	**2678**	5269
B-gpe	12492	12492	12424	12492
I-gpe	154	152	150	154
B-art	319	318	295	319
I-art	252	239	230	252
B-eve	240	240	235	240
I-eve	200	197	195	200
B-nat	162	162	130	162
I-nat	34	34	34	34
Total	839815	580156	730983	873599

Dari Tabel 1 dapat terlihat bahwa terdapat beberapa anomali pada data hasil PF dan EOS, di mana pada PF terjadi penurunan jumlah data dengan tag B-geo sebanyak 3099 data. Hal ini dapat dikatakan sebagai anomali dikarenakan entitas B-geo tidak mungkin direpresentasikan dengan tanda baca (*punctuation*). Tidak hanya B-geo, B-per, B-org, I-org, B-tim dan I-tim juga mengalami anomali data, sehingga memerlukan investigasi lebih lanjut tentang data-data yang terfilter oleh metode PF ini. Sedangkan pada data luaran EOS, jumlah entitas I-geo masih kurang dari jumlah entitas B-geo, di mana hal ini tidak memenuhi asumsi dari metode yang dikembangkan, yaitu jumlah entitas I-geo ≥ B-geo. Jika dilihat dari data pada Tabel 1, maka terdapat sedikitnya 1.201 entitas B-geo yang masih berdiri sendiri, yang seharusnya sudah mendapat pasangan saat proses EOS.

Setelah mendapatkan data *train* luaran dari proses SWF, PF dan EOS, selanjutnya dilakukan *training* terhadap model klasifikasi. Dikarenakan pada uji coba ini menggunakan dua model klasifikasi yang berbeda, maka setelah model pertama selesai dilatih, layer *embedding*-nya akan digunakan kembali pada saat proses *training* model kedua. Dari layer *embedding* tersebut,

embedding matrix hasil perhitungan pada proses train model pertama akan digunakan sebagai bobot matriks dari setiap kata pada proses *training* model kedua. Hal ini dilakukan untuk mendapatkan hasil yang konsisten antara model pertama dan kedua dengan menggunakan data train yang sama. Setelah proses *train* selesai, maka dilakukan evaluasi terhadap kedua model yang dibangun dengan menggunakan data *test* yang tidak melalui proses *resampling*. Hasil evaluasi kedua model dapat dilihat pada Tabel 2 dan Tabel 3.

Tabel 2. Skor Evaluasi model BiLSTM + LSTM

BiLSTM + LSTM	Accuracy	Precision	Recall	F1-score
Baseline	**94.39%**	**38.52%**	**63.85%**	**44.64%**
SWF	83.29%	24.66%	63.57%	30.71%
PF	89.95%	28.95%	60.96%	34.82%
EOS	93.30%	35.49%	60.46%	39.18%

Tabel 3. Skor Evaluasi model BiGRU + LSTM

BiGRU + LSTM	Accuracy	Precision	Recall	F1-score
Baseline	**94.45%**	**36.93%**	63.86%	**43.16%**
SWF	79.31%	22.93%	**65.55%**	28.81%
PF	89.38%	26.22%	58.06%	32.02%
EOS	93.47%	34.84%	63.83%	39.71%

Dari hasil evaluasi dapat dilihat bahwa data train yang melalui proses resampling belum berhasil meningkatkan performa model klasifikasi yang dibangun. Performa model dengan menggunakan data tanpa melalui proses resampling masih lebih baik. Hal ini disebabkan oleh struktur kalimat pada data test tidak lagi sama dengan struktur kalimat pada data train. Unsur acak yang terdapat pada proses EOS membuat beberapa kata yang sama dengan tag B-geo memiliki pasangan kata yang berbeda dengan tag I-geo. Seperti terlihat pada Gambar 1, data setelah EOS memasangkan “London” dengan “Province” dan “Iraq” dengan “Island”. Pada kalimat-kalimat selanjutnya, mungkin saja kata “London” dipasangkan dengan kata lain selain “Province”. Sementara, pada data *test*, kata “London” tetap berdiri sendiri. Hal ini mempengaruhi proses training model klasifikasi yang bersifat *sequence-to-sequence*, dimana proses update bobot memperhitungkan kata sebelum dan sesudah.

Resampling pada level *embedding* memerlukan sumber data komputasi yang besar, mengingat data yang awalnya berupa angka tunggal, diubah menjadi vektor dengan dimensi tinggi. Jika ingin mengurangi dimensi vektor agar kompleksitas komputasi berkurang, dapat berakibat pada menurunnya performa model klasifikasi. Hal ini disebabkan oleh ruang vektor untuk representasi kata menjadi lebih terbatas, sehingga kata-kata yang seharusnya berjauhan secara jarak, menjadi lebih dekat dibandingkan jika dimensi vektor yang digunakan tinggi. Perlu dilakukan analisis lebih lanjut untuk mencari titik tengah antara dimensi vektor, performa model klasifikasi dan kompleksitas komputasi agar dapat melakukan *resampling* pada level representasi (embedding) kata.

KESIMPULAN

Metode *Entity-based Oversampling* (EOS) menghasilkan dataset dengan komposisi yang tidak bisa diprediksi rationya, dikarenakan sangat bergantung pada jumlah kata dengan tag B yang berdiri sendiri. Jika pada dataset tidak terdapat kata dengan tag B yang beridiri sendiri, maka metode ini tidak akan melakukan apa-apa terhadap dataset. Setelah melakukan uji coba, dataset yang dihasilkan dari proses *oversampling* belum berhasil meningkatkan performa model klasifikasi yang dibangun. Selain itu, terdapat beberapa anomali pada dataset luaran dari proses oversampling yang memerlukan investigasi lebih lanjut, khususnya pada metode yang diusulkan. Metode yang diusulkan masih mengandung usur "acak" yang membuat luaran dari teknik ini sulit untuk diprediksi. Perlu dilakukan pengembangan lebih lanjut agar proses pemilihan kandidat kata dan pembangkitan kata bisa dilakukan secara heuristik, sehingga luaran akan lebih dapat dikontrol.

Percobaan untuk melakukan *oversampling* pada level representasi (*word embedding*) kata masih perlu ditinjau ulang dari sisi dimensi vektor dan ukuran *padding* yang digunakan agar dapat mengoptimalkan penggunaan sumber daya komputasi yang akan dibutuhkan saat proses *oversampling*. Skor evaluasi *recall*, *precision* dan *F1* yang rendah dipengaruhi oleh dataset yang bersifat *well-annotated*, dimana terdapat 17 kelas. Sehingga semakin tinggi kemungkinan terjadinya *missclassify* serta proses perhitungan skor *recall*, *precision* dan *F1* memiliki banyak pembagi (macro-average). Kode sumber penelitian ini dapat diakses melalui github[2].

DAFTAR PUSTAKA

[1] A. M. Gliozzo, C. Giuliano, and R. Rinaldi, "Instance Pruning by Filtering Uninformative Words: An Information Extraction Case Study," in *Computational Linguistics and Intelligent Text Processing, 6th International Conference, CICLing 2005, Mexico City, Mexico, February 13-19, 2005, Proceedings*, 2005, vol. 3406, pp. 498–509, doi: 10.1007/978-3-540-30586-6_54.

[2] A. Akkasi, E. Varoğlu, and N. Dimililer, "Balanced undersampling: a novel sentence-based undersampling method to improve recognition of named entities in chemical and biomedical text," *Appl. Intell.*, vol. 48, no. 8, pp. 1965–1978, 2018, doi: 10.1007/s10489-017-0920-5.

[3] A. Akkasi, "Sentence-based undersampling for named entity recognition using genetic algorithm," *Iran J. Comput. Sci.*, vol. 1, no. 3, pp. 165–174, 2018, doi: 10.1007/s42044-018-0014-5.

[4] A. Akkasi and E. Varoglu, "Improvement of Chemical Named Entity Recognition through Sentence-based Random Under-sampling and Classifier Combination," *J. AI Data Min.*, vol. 7, no. 2, pp. 311–319, 2019, doi: 10.22044/jadm.2018.5929.1700.

[5] J. Li, A. Sun, J. Han, and C. Li, "A Survey on Deep Learning for Named Entity Recognition," *CoRR*, vol. abs/1812.0, 2018, [Online]. Available: http://arxiv.org/abs/1812.09449.

[2] https://github.com/swardiantara/fp-kk-2021