Identifying Support for Nombank Predicates

Aashka Trivedi
NYU Courant
aht323@nyu.edu

Sarvani Nadiminty
NYU Courant
sn2884@nyu.edu

Raksha Hegde
NYU Courant
rh3468@nyu.edu

Abstract

This work aims at identifying the support for nominal predicates of the NomBank Database (Meyers et al., 2004). We experiment with various features, such as word-related features, sentence-level features, and grammatical features to improve the performance of a maximum entropy model to classify whether each word of the sentence is the support for a given predicate. Our best system results in a 24 point improvement on the F1-score from our baseline system.

1 Introduction

As semantic relations are distinct we use a unique way of describing the relations called predicate-argument structure. A Predicate (often verb) needs Arguments to bring meaning to a sentence. Generally a nominal predicate is a noun that expresses properties about the subject or object. Nominal predicates are composed by a support verb like give, by, to be, so on and a predicate (PRED), which is the predicate in a sentence and has arguments. The support verb only provides to the Predicate in the sentence. Support verbs occur in Support chains along with transparent nouns. They can be considered similar to light verbs such as make, give, take, and other empty verbs. They help in nominalizing the predicate.

Below is an example of how a sentence in the dataset looks like. The sentence being "This small Dallas suburb's got trouble".

This	DT	B-NP	0	
small	JJ	I-NP	1	
Dallas	NNP	I-NP	2	
suburb	NN	I-NP	3	ARG1
's	VBZ	O	4	
got	VBN	O	5	SUPPORT
trouble	NN	B-NP	6	PRED

The correct identification of the arguments and support verbs of a nominal predicate is important to the task of Semantic Role Labeling (SRL). The entire approach of this paper incorporates striving for enabling every word in a sentence to become effective and supportive for the predicate so as to provide more insightful outcomes. This paper describes the previously done research about the support verbs for nominal predicates. The intention of this paper lies in focusing on efforts in terms of generating an optimum NomBank-centric SUPPORT verb tagger. We first will set an upper bound on our system's performance by assuming it knows both the predicate and its argument(s), and then we'll utilize the feature data generated to predict SUPPORT words. Next we set a lower bound on our system's performance by assuming it knows only the predicate and then we'll utilize it's feature data to predict ARGUMENTS and SUPPORT words. In this paper we discuss different kinds of models that we have experimented with (around 6) and compare them using performance metrices like Precision, Recall and F-1 Score. Our work aims at building a system that can be trained to automatically identify the Support words. The training data consists of predicate and support annotations. We add features such as POS, distance from the predicate, forward distance, backward distance, BIO-tags, etc. We also use the most frequently used Support words list and a transparent noun list for this task.

All the code, features and models for this work can be found on Github.[1]

2 Related Work

Since the work involving the identification of support words for Nombank is relatively sparse, we hope to provide insights into feature selection. However, we draw inspiration from prior work involving the automatic identification of *light verbs* or *support verbs* (Teufel and Grefenstette, 1995; Dras, 1995; Nagy et al., 2011; Chen et al.,

[1] https://github.com/aashka-trivedi/nombank-support-identification

2015), which lack semantic meaning in themselves, but are usually ejected from the original verb's syntactic role during the nominalization. Halliday's(Halliday, 1985) notion of *Lexical Density* where each word carries different semantic weight has been the base to identify the semantic lightness of any word. For example, the verb *propose* is nominalized to *make a proposal*, where *make* is the corresponding support or light verb making it semantically light. Our model will use light verbs with Noun Phrase complements only, by studying the link between a verb's relative frequency in these light verb expressions and its semantic lightness.

Grefenstette and Teufel (Teufel and Grefenstette, 1995) agree that the support verb to be used with a nominalized predicate structure is unpredictable. However, they introduce a method of deriving probable support verbs given a verb and its nominalization, using POS tagging, syntactic analysis using a robust parser and frequency statistics over a corpora. Taking all verb-object pairs (object is deverbal nominalisation) is one technique to extract light verbs from a corpus. Grefenstette and Teufel (Teufel and Grefenstette, 1995) use only Local Information (specific nominal)- when all the occurrences of noun in verb-object pairs are counted, a local relative frequency for each verb is calculated with regard to that noun. Dras (Dras, 1995) uses Local information with Global information. A global relative frequency for each verb is calculated by counting all instances of that verb, independent of their objects. Both local and global information are integrated to generate a changed likelihood of becoming a support verb. His model revealed some information on how sometimes there can be more than one proposed support verb. For ex: *cause/do harm*. Here both *cause* and *do* can be a good choice for a support verb and their frequency ratio is quite small (Frequency ratio = ratio of product of local and global relative frequencies for the first choice to the product of local and global relative frequencies second choices). This is true for many other support verbs like *bear/have resemblance* or *make/produce change*. It is also apparent that the relative frequency of a verb in light verb constructions appears to be a very reasonable predictor of a verb's supporting behaviour, resulting in viable options for support verbs for nominalizations.

Vincze et.al in (Nagy et al., 2011) uses rule-based and Machine Learning based approaches to identify Light Verb Constructions(LVC) from Multiword expressions in different domains. POS-rule method accepts the n-grams for which the predefined patterns such as VB.? (NN|NNS) could be applied as LVC. The 'Most Frequent Verb'(MFV) method uses the fact that most of the light verbs are the most frequently occurring verbs (e.g. do, make, take etc.). Various other methods such as 'Suffix','Stem' etc. are combined with the ones above in the rule based approach. The MFV method and the intersection of MFV and 'Stem' was found to be the most useful rule based approaches. In the Machine Learning approach, a Conditional Random Field(CRF) classifier is used along with features - domain specific dictionaries of first names, company types, token frequency, POS tags and contextual information such as sentence position, trigger words, etc. CRF classifier yields better results when used with an extended feature set that includes the rule-based methods as LVC specific features. The features used for LVC identification in (Chen et al., 2015) are POS tags of the support verb(Vs) and the Noun(N) pairs and the surrounding words, the dependency tree node distance between Vs and N, the voice (active/passive) of the Vs and the word sense tags from OntoNotes and WordNet.

3 Data

Our data comes from the cleaned version of Nombank (Meyers et al., 2004), which we process by extracting sentences that contain at least one SUPPORT tag. We use all nominal predicates for this task, resulting in about 2000 instances of support words in the training set, 1000 instances of support in the test set and 600 instances of support in the development set.

Each entry in Nombank consists of the word itself, the POS tag, the BIO tag and the role label (such as the predicate (PRED), the arguments (ARG0- ARG4), and the support (SUPPORT). The cleaned version also assumes one predicate per sentence, so a sentence with two predicates would be split into two separate entries, each corresponding to one predicate. For example, the sentence *They expect a 5% to 10% increase in gas prices.* has two entries in our dataset, with two separate predicates: *They expect a 5% (PRED) to 10% increase in gas prices.* and *They expect a 5% to 10% (PRED) increase in gas prices.*

In our dataset, there may be multiple instances

of SUPPORT in a sentence, for example: *A breakdown showed that food prices \ARG1 were the most active part of growth with \SUPPORT a rise of 0.6 % \PRED* . Moreover, SUPPORT can occur for any type of Argument, however it occurs most frequently in the context of ARG1.

4 Methodology

Our work focuses on the use of intelligently crafted features to improve the precision, recall and F1 score of our model in order to classify support words for nombank predicates. For the same, we prepare a base model, which uses only *word-related features*, such as the stem of the word, POS tag, BIO tag, etc. We then increase our feature set to include more non-trivial features, as described in the next section, such as grammatical features and distance related features.

The concept of a support word is always in context of a specific predicate and an argument. Keeping this in mind, we experiment with two levels of knowledge to the system:

1. Predicate Knowledge: here we assume that the system has the minimum required knowledge, that is, knowledge of which word is the predicate for which the support is being found. This system would include features indicating whether the specific word is the predicate, the distance of the word from the predicate, etc.

2. Predicate and Argument Knowledge: here, the system knows which words are the predicates, and which words are the arguments (and which type of argument the word is). This system would include features like the type of argument, distance from ARG1, etc.

We then create one model for each feature set, and for each level of knowledge. Here, we use a Maximum Entropy model - GIS from Java's OpenNLP Package[2]. The model is trained iteratively by streaming in the features.

5 Experiments

We experiment with the following types of features:

1. Word Related Features: these are properties of the word itself. Features used are word, POS tag, BIO Tag, Stem of word (Here, we stem the word using NLTK's Porter Stemmer[3]).

2. Sentence Related Features: these features incorporate the knowledge of previous and following words. The features include all the word-related features for a 5 word window (i.e., two words before and two words after each word).

3. Role Related Features: these features encode the semantic role (PRED, ARG0, ARG1, ARG2, ARG3, ARG4) of the word in the sentence, depending on what level of knowledge the system has. The feature is_predicate is always present, while the system gets the feature is_argument (for each type of argument from ARG0-ARG4) for cases when the system is given knowledge about the arguments.

4. Distance Related Features: we encode the distance of a word from another word having a specific role-label, as this may be helpful in identifying support (for example, most SUPPORT words lie near the arguments or predicates). These features include forward and backward distance from predicate, whether word lies in a 5 word window of each argument (again, the latter feature is only present in systems with the knowledge of both Predicates and Arguments).

5. Transparent Nouns: Transparent nouns typically occur near support words. We obtain a list of transparent nouns in Nombank (please refer to Appendix A for this list), and include a feature for if the word (or stem of the word) is in the list of transparent nouns, or lies in a 7 word window of a transparent noun.

6. Support Verb: Support Verbs are semantically empty verbs that are ejected when a verb is nominalized. We include a feature if the word is in the list of support verbs (please refer to Appendix B), as many support verbs are also tagged as support.

7. Previous Tag: we encode whether the previous word was predicted to be a support word. This is helpful in cases like *The price increased by 5%*, where both *increased* and *by* should be tagged as support words.

[2] http://maxent.sourceforge.net/about.html

[3] https://www.nltk.org/_modules/nltk/stem/porter.html

Model	System Knowledge	Feature Set
Model 0	Predicates	Word, Sentence
Model 0 Arg	Predicates, Arguments	Word, Sentence
Model 1	Predicates	Word, Sentence, Distance
Model 1 Arg	Predicates, Arguments	Word, Sentence, Distance
Model 2	Predicates	Word, Sentence, Distance, Transparent Nouns
Model 2 Arg	Predicates, Arguments	Word, Sentence, Distance, Transparent Nouns
Model 3	Predicates	Word, Sentence, Distance, Transparent Nouns, Support Verbs
Model 3 Arg	Predicates, Arguments	Word, Sentence, Distance, Transparent Nouns, Support Verbs
Model 4	Predicates	Word, Sentence, Distance, Transparent Nouns, Support Verbs, Previous Tag
Model 4 Arg	Predicates, Arguments	Word, Sentence, Distance, Transparent Nouns, Support Verbs, Previous Tag
Model 5	Predicates	Word, Sentence, Distance, Previous Tag
Model 5 Arg	Predicates, Arguments	Word, Sentence, Distance, Previous Tag

Table 1: Model Descriptions. Each model corresponds to a level of knowledge, and a specific feature set.

We build a total of twelve models - six models for each knowledge level (Predicate known and Predicate and arguments known), and incrementally adding features to each model. The models are described in Table 1.

6 Results

The results for each model discussed in the previous section on the test set are summarized in Table 2. In terms of Recall and F1 score, Model 1 Args (with word, sentence and distance related features) gives the best performance with respect to the Base Model, Model 0. Specifically, we see a 27 point jump in precision, 19 point jump in recall and 24 point jump in F1-Score. In terms of precision, Model 4 Args (with word, sentence, distance, transparent noun, support verb and previous tag features) has the maximum gain in performance, with a 31 point increase in precision.

7 Discussion

We make the following observations from the results in Table 2:

1. Giving the system knowledge of all arguments help improve performance for every set of features. This is intuitive, as the system is aware of which argument's support it is looking for.

2. Integrating grammatical features (like knowledge of transparent nouns and support verbs) helps improve performance, but only by a small percent.

3. While our models are able to reach higher levels of precision (0.73), we are not getting as big a jump in recall. This is because our model only predicts a few words of the corpus to be support, but among those predicted, it gets a high percentage of them correct.

One way to improve recall may be to use more training data. Currently, there are a little under 2000 instances of support in the training data. Collecting more instances of support for nominal predicates may be an interesting future direction of this work. Moreover, a more comprehensive list of support verbs and transparent nouns may help further improve performance.

Another observation is that the model is not predicting the SUPPORT words based on the position of the PREDICATE. For example in this sentence: "Morevover, this year's good inflation news many have ended last month, when energy prices zoomed up 6.5% after plunging 7.3% in August".Here the ARG is Prices. If 6.5 "%" is the PRED then SUPPORT should be "zoomed". and if 7.3 "%" is the PRED then SUPPORT should be "plunging". But our model is either tagging both the words as SUPPORT or none of the words as SUPPORT.

The models are also not identifying words like "increase by", "rose by", "decline by". Here the "word by" is not being tagged as SUPPORT and is being ignored.

8 Conclusion

This paper presented a system to predict the SUPPORT words given a nominal predicate. One of

Model	Number of Predicted Support	Precision	Recall	F1 Score
Model 0	475	0.42	0.13	0.20
Model 0 Arg	452	0.42	0.13	0.20
Model 1	659	0.64	0.28	0.39
Model 1 Arg	683	0.69	0.32	0.44
Model 2	615	0.65	0.27	0.38
Model 2 Arg	645	0.71	0.31	0.43
Model 3	617	0.65	0.28	0.39
Model 3 Arg	647	0.71	0.31	0.43
Model 4	479	0.68	0.22	0.33
Model 4 Arg	515	0.73	0.25	0.38
Model 5	522	0.66	0.23	0.34
Model 5 Arg	545	0.72	0.26	0.39

Table 2: Performance of Models on the test set. The test set had 1492 support words in the key. The suffix "Arg" denotes that the model has knowledge of the predicates and all arguments, an omitted suffix denotes the system has knowledge of only predicates. A detailed description of the features of each model is given in Table 1.

the common criticisms of semantic roles, and presumably a prime drawback toward their adoption in NLP, could be assumed as their restricted coverage. As the hand-built semantic role tagging costs a lot, it is merely conceivable that the anticipated quality annotation would ultimately be gained for each and every predicate of English. Furthermore, the lexically distinct norm of the mapping between semantic roles and surface syntax makes it troublesome to marginalize it from perceived predicates to unseen predicates and therefore, no training data is currently prevailing. Mechanisms for expanding the SRL coverage strongly address a significant need. The intention of this study lies in focusing on generating a well working model as part of the core procedure in relation to how techniques applied in creating this SRL system is able to be transferred to propping up NomBank SRL system while it is also significant to keep an eye on making NomBank-specific enlargements and enhancements to the core baseline system.

9 Future Works

This paper presented initial steps towards creating a system to identify support for Nombank arguments, however, we feel that there is a large potential to further improve this work. As noted, the biggest drawback of this system was the size of training data, and so it may be beneficial for more data with support word annotations to be collected and used for training the system further. The current system could also be enhanced to jointly predict arguments along with the Support words by updating the training data to include the ARG labels.

Another promising future directions includes fine-tuning large language models pretrained on masked language modelling tasks, such as BERT (Devlin et al., 2019), for this task. BERT's prime technique is implementing the bi-directional training of data, a popularized system with attention to language modeling. A language model that is prepared by means of being bidirectionally trained can contain a deeper sense of language flow and context than single-direction language models. Fine-tuning BERT to label the semantic roles of each word for the sentence may be exploited for Support labelling, and may be straightforward with a large training corpus.

References

Wei-Te Chen, Claire Bonial, and Martha Palmer. 2015. English light verb construction identification using lexical knowledge. In *AAAI*, pages 2368–2374.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mark Dras. 1995. Automatic identification of support verbs: A step towards a definition of semantic weight. In *Proceedings of the Eighth Australian Joint Confer-*

ence on Artificial Intelligence, pages 451–458, Singapore. World Scientific Publishing Co. Pte.

Michael Alexander Kirkwood Halliday. 1985. Spoken and written language. Waurn Ponds, Vic. : Deakin University : distributed by Deakin University Press.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: An interim report. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, pages 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

István T. Nagy, Gábor Berend, György Móra, and Veronika Vincze. 2011. Domain-dependent detection of light verb constructions. In *Proceedings of the Second Student Research Workshop associated with RANLP 2011*, pages 1–8, Hissar, Bulgaria. Association for Computational Linguistics.

Simone Teufel and Gregory Grefenstette. 1995. Corpus-based method for automatic identification of support verbs for nominalizations. In *Seventh Conference of the European Chapter of the Association for Computational Linguistics*, Dublin, Ireland. Association for Computational Linguistics.

A List of Transparent Nouns

The following words are treated as transparent nouns:

%, 1/10th, abundance, acre, act, afternoon, amount, amp, anthology, army, array, arsenal, assembly, assortment, atom, avalanche, babel, bag, band, barrage, barrel, base, basket, batch, battalion, battery, bestiary, bevy, billions, binge, bit, blanket, blink, blitz, bloc, block, board, boatload, body, book, bottle, bout, bowl, box, branch, brand, breed, briefcase, brigade, brotherhood, bucket, buffet, bulk, bunch, bundle, burst, bushel, busload, buzzsaw, cabinet, cache, cake, camp, can, capsule, carat, cargo, cartel, carton, cascade, case, cast, category, cent, chamber, chest, chip, chorus, chunk, circle, clan, class, clique, cloud, club, clump, cluster, clutch, clutter, cm, coalition, column, committee, community, compendium, complex, conglomerate, consortium, contingent, convoy, core, cornucopia, corps, council, couple, covert, crew, crop, crowd, crush, cult, cup, dash, dearth, decimeter, deck, deluge, dollar, dose, dozen, draft, dram, dribble, drop, drumroll, edition, eighth, element, end, episode, epsiode, faculty, family, family, feeler, fiber, field, fifth, finger, fit, flat, fleet, flock, flotilla, flurry, folio, foot, force, form, fortune, fourth, foyer, fraction, franc, gallery, gallon, gamut, genre, glade, glimmer, glut, grain, gram, group, group, grove, guilder, hail, half, handful, hank, head, heap, herd, hill, hoard, hodgepodge, hole, horde, host, house, howl, hundreds, hunk, inch, instance, interest, interest, iota, jillion, jillions, jungle, kernel, kilobyte, kilogram, kilometer, kind, layer, league, legion, letter, library, library, line, line, line, lira, litany, load, loaf, lobby, lot, machine, majority, mass, mass, matter, maze, measure, megabyte, megabytes, megawatt, menu, mess, meter, mile, million, minority, minute, mob, modicum, molecule, month, mound, mountain, mu, multitude, neighborhood, network, night, nothing, number, oasis, office, onslaught, opening, organization, other, ounce, oversupply, pack, pack, package, packet, page, pageant, pair, panel, parade, parody, part, party, passel, patch, peck, percent, percentage, peseta, phalanx, piece, pile, pint, platoon, plenty, plethora, plume, plurality, point, pool, portfolio, portion, pot, potpourri, pound, queue, quota, raft, rainbow, range, rash, ream, remainder, remnant, rendition, replica, reservoir, residue, rest, roll, roomful, roster, round, row, run, rush, sack, sample, score, second, sequence, series, series, set, share, shareholding, sheaf, sheet, shred, slate, slew, slice, slice, slip, sliver, smattering, smidgin, smidgins, sort, spate, species, spectrum, squad, squadron, stable, stack, staff, stake, steering, stockpile, storm, story, strain, streak, stretch, string, strip, stuff, subcommittee, subset, sum, surfeit, swarm, swath, sweep, swirl, symphony, syndicate, tangle, team, teaspoon, tens, tenth, thicket, third, thousands, tidbit, tie, tin, tinge, ton, torrent, touch, tract, trail, trillion, trove, trust, tube, two-hundredth, type, union, unit, unit, variety, variety, vein, version, vial, volley, volume, wad, warren, watt, wave, wealth, web, welter, whirl, whirlwind, whole, worth, yen, coil, jumble, pinpoint, quart, ribbon, shelf

B List of Support Verbs

The following words are treated as support verbs:

get, revive, level, heed, commit, go-along-with, adhere-to, begin, compete-for, frame, press, aim-at, force, renew, subject, flood, seek, take-part-in, listen-to, play, decide-about, base-on, pass, slate, escalate, bear, contest, circumvent, flout, launch, shift-to, withhold, follow, furnish, produce, receive, vote-for, dissociate, act-on, implicate, modify, negotiate, require, recant, toss-out, evade, provide, put-on, place-on, yank, guarantee, reject, appreciate, orchestrate, plot, arrange, spurn, avoid, botch, undergo, lob, pursue, undertake, break-off, obtain, revoke, elicit, bungle, best, extract, disregard,

have, unleash, cancel, firm-up, join-in, premeditate, share, smart-from, post, deny, attract, accelerate, hurl, rebut, start, base+on, wreak, sing, fight, conduct, retract, continue, continue-with, intensify, charge, invite, prepare, replicate, step-up, fail, give, offer, sift-through, welcome, proclaim, face, march-to, absolve, return-with, replace, trade, swamp, ply, further, approve, credit, hear, study, exchange, laugh-off, explore, fend-off, acquit, approach, balk-at, decide, protect, allow, augment, blurt-out, kowtow-to, consign, dismiss, spearhead, take, brush-aside, reconsider, thwart, mount, persevere-with, confess-to, heap-on, decline, need, fling, soften, abide-by, prepare-for, assign, weigh, match, carry-out, get-away-with, clear, call-for, obey, rescue, inundate, have_in, answer, escape, lead, come-from, comply-with, execute, lay-on, fulfill, stage, propose, besiege, choose, disprove, cause, parcel-out, scoff-at, engineer, draft, attempt, follow-with, pepper, reverse, induce, perform, amend, woo, intrigue, disobey, entertain, perpetrate, arraign, accuse, deluge, attach-to, counter-with, complete, boost, do, flood-with, intend, restructure, issue, hold-down, end, dispute, pass-up, renege-on, levy, address, carry, respond-to, deluge-with, satisfy, concentrate, levy-at, submit, suspend, spread, succeed-in, rescind, save, file, enhance, initiate, offend-with, take-advantage-of, give-up, inflict, refuse, collaborate-in, shoulder, get-in, pin-on, cease, affix-on, engage-in, bombard, carry+out, jack-up, calculate, apply-for, counter, organize, sign, take-up, put-on-the-table, rebuff, single-out, defy, cast-on, stand-for, plan, jump-at, withdraw, increase, wage, grant, couple-with, admit, present, convict, react-to, license, fabricate, agree-to, come-back-with, risk, deliver, suffer, sweeten, turn-down, come-forward-with, consider, go-over, abort, get-out, ease-up-on, hit-with, mark, coerce, gain, mull-over, design, postpone, make, extend, participate-in, drop, bring, resort-to, assume, shepherd, tender, take-back, reciprocate-with, win, violate, miss, accept, ignore, resist