System rozpoznawania mowy polskiej dla robota społecznego

Artur Zygadło
Politechnika Warszawska, Wydział Mechaniczny Energetyki i Lotnictwa, ul. Nowowiejska 24, 00-665 Warszawa

Artur Janicki
Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, Instytut Telekomunikacji, ul. Nowowiejska 15/19, 00-665 Warszawa

Przemysław Dąbek
Przemysłowy Instytut Automatyki i Pomiarów PIAP, Al. Jerozolimskie 202, 02-486 Warszawa

Streszczenie: W artykule przedstawiono system automatycznego rozpoznawania mowy polskiej dedykowany dla robota społecznego. System oparty jest na bezpłatnej i otwartej bibliotece oprogramowania *pocketsphinx* (CMU Sphinx). Przygotowano zbiory nagrań: treningowy i testowy wraz z transkrypcjami. Zbiór treningowy obejmował głosy 10 kobiet i 10 mężczyzn i został przygotowany na podstawie audiobooków, natomiast zbiór testowy – głosy 3 kobiet i 3 mężczyzn nagrane w warunkach laboratoryjnych specjalnie na potrzeby pracy. Przygotowany zbiór fonemów dla języka polskiego, składający się z 39 fonemów, opracowany został na podstawie dwóch popularnych zbiorów dostępnych danych. Słownik fonetyczny opracowano za pomocą funkcjonalności konwersji grapheme-to-phoneme z biblioteki eSpeak. Model statystyczny języka dla tekstu referencyjnego składającego się z 76 komend wygenerowano za pomocą programu *cmuclmtk* (CMU Sphinx). Uczenie modelu akustycznego oraz test jakości rozpoznawania mowy przeprowadzono za pomocą programu *sphinxtrain* (CMU Sphinx). W warunkach laboratoryjnych uzyskano wskaźnik błędu rozpoznawania słów (WER) na poziomie 4% i błędu rozpoznawania zdań (SER) na poziomie 9%. Przeprowadzono też badania systemu w warunkach rzeczywistych na grupie testowej złożonej z 2 kobiet i 3 mężczyzn, uzyskując wstępne wyniki rozpoznawania na poziomie 10% (SER) z bliskiej odległości oraz 60% (SER) z odległości 3 m. Określono kierunki dalszych prac.

Słowa kluczowe: automatyczne rozpoznawanie mowy, command and control, robot społeczny

1. Wprowadzenie

Naturalnym sposobem komunikacji międzyludzkiej jest komunikacja werbalna, dlatego w kontekście interakcji człowiek–maszyna dąży się do opracowywania systemów automatycznego rozpoznawania mowy (ARM). Funkcjonalność ta jest szczególnie ważna w przypadku robotów społecznych [1]. Roboty społeczne to roboty przeznaczone do działania razem z człowiekiem w jego codziennym otoczeniu, przy czym ich cechą charakterystyczną jest komunikowanie się z człowiekiem za pomocą sygnałów werbalnych i niewerbalnych. Zagadnienie automatycznego rozpoznawania mowy dotyczy nie tylko robotów społecznych [2–5], ale m.in. również kontrolowania trajektorii ruchu manipulatorów przemysłowych [6, 7] lub pojazdów bezzałogowych [8].

Proces automatycznego rozpoznawania mowy polega na zamianie mowy ludzkiej zarejestrowanej przez mikrofon na tekst. Obecnie do rozwiązania tego zagadnienia najczęściej wykorzystuje się metody oparte na statystycznym rozpoznawaniu wzorców z użyciem tzw. niejawnych modeli Markowa HMM (ang. *Hidden Markov Models*). Niejawne modele Markowa pozwalają określić najbardziej prawdopodobną sekwencję kolejnych stanów nieobserwowalnego procesu na podstawie sekwencji obserwacji cechujących się pewną wariancją. W przypadku systemów ARM stany procesu mogą być fonemami, czyli elementami z pewnego skończonego zbioru, jakie fonologia wyróżnia w sygnałach dźwiękowych wszystkich wypowiedzi w danym języku. Obserwacjami natomiast są pewne charakterystyczne cechy ekstrahowane z kolejnych segmentów czasowych sygnału dźwiękowego konkretnej wypowiedzi, którą chcemy zamienić na tekst. Znane są także próby stosowania metod sztucznej inteligencji w rozpoznawaniu mowy, a konkretnie sztucznych sieci neuronowych [9].

Systemy ARM dzielimy w zależności od charakteru planowanego zastosowania na systemy typu:

Autor korespondujący:
Przemysław Dąbek, pdabek@piap.pl

Artykuł recenzowany
nadesłany 09.08.2016 r., przyjęty do druku 21.11.2016 r.

— *command & control* – rozpoznaje tylko komendy z określonego wcześniej zbioru,
— *continuous speech recognition* – rozpoznaje dowolne wypowiedzi mające sens w danym języku.

Systemy typu *command & control* są mniej wymagające, zarówno co do niezbędnej ilości danych treningowych, jak i zasobów sprzętowych niezbędnych do pracy dekodera.

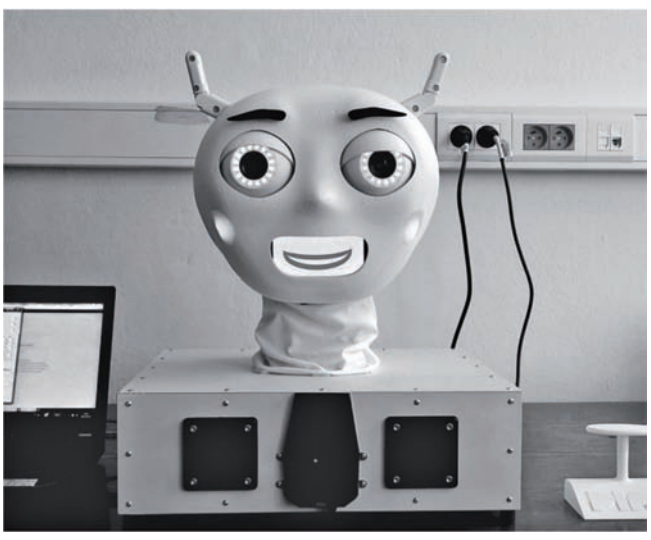
Istniejące oprogramowanie implementujące algorytmy automatycznego rozpoznawania mowy można podzielić ze względu na rodzaj licencji i dostępność kodu źródłowego na dwie grupy: programy komercyjne i pakiety oprogramowania typu *open-source*. Rozwiązania komercyjne zapewniają wysoką jakość rozpoznawania mowy, jednak nie są dostępne dla wszystkich języków oraz wymagają znacznych nakładów finansowych. Ponadto nie zawsze są one wystarczająco elastyczne, aby spełnić wymagania projektowe. W szczególności wadą tego typu rozwiązań jest brak możliwości opracowania własnych komponentów systemu ARM, zwłaszcza modelu akustycznego. Pakiety oprogramowania z ogólnodostępnym kodem źródłowym, do których zaliczają się m.in. *HTK* [16], *CMU Sphinx* [17] oraz *Kaldi* [18] zawierają gotowe modele dla wielu języków, jednak rozpoznawanie mowy w języku polskim wymaga opracowania własnych komponentów – modelu akustycznego, słownika fonetycznego oraz modelu statystycznego języka.

Praca [7] dotyczy rozpoznawania mowy w języku polskim, jednak autor używa oprogramowania *Microsoft SAPI*, które nie pozwala tworzyć własnego modelu akustycznego. W pracy [10] wykorzystano oprogramowanie *CMU Sphinx* na potrzeby języka polskiego, poza dziedziną robotyki do sterowania głosowego grą komputerową, natomiast w publikacjach [11, 12] zastosowano pakiet *Kaldi*.

Celem niniejszej pracy jest przedstawienie systemu automatycznego rozpoznawania mowy opracowanego na potrzeby sterowania robotem społecznym oraz wyników wstępnych badań. Zakres opracowania obejmował dobór modułów oprogramowania, przygotowanie danych do treningu modelu akustycznego, przygotowanie słownika fonetycznego i modelu języka oraz przygotowanie danych testowych.

Opracowany system spełnia następujące założenia:
— jest systemem typu *command & control* przeznaczonym dla robota społecznego,
— jest dedykowany dla języka polskiego,
— ma rozpoznawać mowę dowolnej osoby dorosłej,
— jest zaimplementowany w oparciu o gotowe rozwiązania typu *open-source*.

Obecny artykuł bazuje na wynikach pracy inżynierskiej [13].



Rys. 1. Prototyp głowy robota społecznego IRYS
Fig. 1. Prototype head of IRYS social robot

2. Opracowanie systemu ARM

2.1. Założenia

Projektując system ARM, zdecydowano się wykorzystać bibliotekę *CMU Sphinx* dla języka C (*pocketsphinx*). Jest to implementacja systemu ARM opracowana głównie do celów badawczych na Carnegie Mellon University w Pittsburghu. Biblioteka była kompilowana z kodów źródłowych pobranych z repozytorium projektu [17] w sierpniu 2015 r. Taki wybór był podyktowany dostępnością materiałów źródłowych [19] oraz wcześniejszymi doświadczeniami wykorzystania biblioteki dla języka polskiego [10]. Rozważano również użycie pakietu *Kaldi*, który jest wskazywany jako bardziej zaawansowany [14]. Pakiet ten był już stosowany do rozpoznawania mowy polskiej w innych badaniach [11, 12]. Pakiet *CMU Sphinx* umożliwia osiągnięcie dobrych rezultatów w krótkim czasie [14] i z tego względu *pocketsphinx* został wybrany do realizacji omawianych prac badawczych.

System ARM jest przeznaczony do pracy z robotem społecznym IRYS opracowywanym w Przemysłowym Instytucie Automatyki i Pomiarów PIAP (rys. 1). System powinien umożliwiać sterowanie poszczególnymi stopniami swobody głowy robota za pomocą wypowiadanych poleceń. Określono listę komend, które powinny być rozpoznawane przez system ARM. Zbiór ten składa się z poleceń związanych z ruchami poszczególnych stopni swobody głowy robota (szyi, oczu, powiek, brwi, uszu) oraz z okazywaniem przez niego emocji (radość, smutek, strach, zaskoczenie, wstręt, złość). Przykładowo, system rozpoznaje komendy *„obróć oko prawe o piętnaście stopni w prawo”* czy *„uśmiechnij się”*. Pełna lista 76 poleceń uznawanych przez system za prawidłowe znajduje się w pracy [13], a wersja kompaktowa listy jest opisana w punkcie 2.6.

2.2. Algorytm postępowania

Sposób postępowania prowadzący do przygotowania systemu do działania przedstawiono na rys. 2. Przygotowanie systemu przebiega w dwóch zasadniczych etapach: (1) uczenie modelu akustycznego, (2) test jakości działania wytrenowanego systemu. Przeprowadzenie testu jest integralną częścią procedury, narzuconą przez moduł trenujący. Przygotowany w ten sposób system można poddać dalszym badaniom.

Jako pierwszy krok, należy przygotować:
— zbiór danych treningowych i zbiór danych testowych,
— zbiór fonemów i słownik tzw. wypełniaczy (cisza oraz dźwięki bez znaczenia, lecz obecne w nagraniach),
— reguły fonetyczne języka dla programu zamieniającego zapis literowy słów na ich zapis fonetyczny,
— konfigurację programu *sphinxtrain*, za pomocą którego przygotowywany jest model akustyczny.

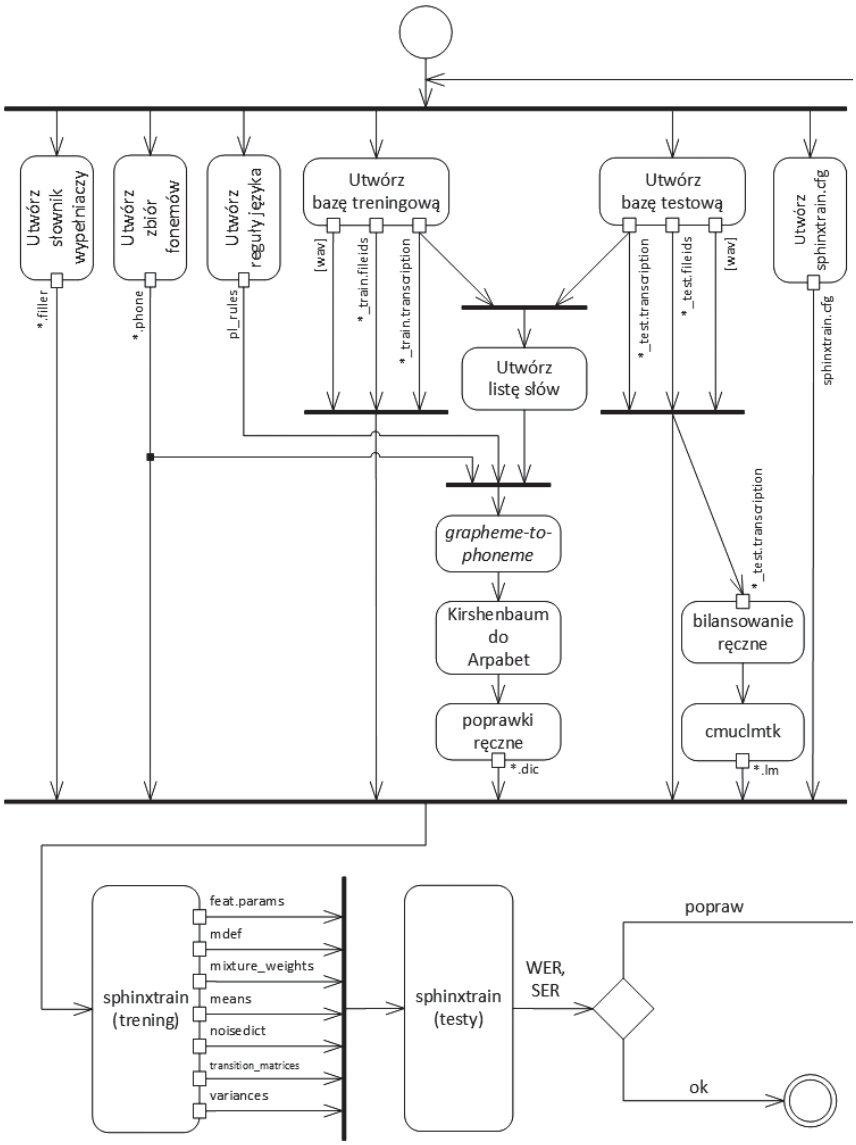
W drugim kroku, należy wygenerować słownik fonetyczny (.dic) oraz model statystyczny języka (.lm).

W trzecim kroku, należy uruchomić program *sphinxtrain*. Wynikiem tego działania jest zbiór plików opisujących model akustyczny oraz wskaźniki jakości dekodowania zbioru testowego za pomocą opracowanego systemu. Jeśli wartości uzyskanych wskaźników nie są zadowalające, można przeprowadzić dostrajanie systemu.

2.3. Zbiór treningowy i testowy

Zbiór treningowy (baza treningowa) to kolekcja nagrań wypowiedzi wraz z ich transkrypcjami. Służy on do wyuczenia modelu akustycznego (trening).

W przypadku systemu przeznaczonego dla dowolnego użytkownika, baza powinna zawierać próbki głosu mówców obu płci i w zróżnicowanym wieku, zarówno wypowiedzi szybkie, jak i powolne. Ponadto zbiór próbek powinien cechować się różnorodnością w występowaniu sekwencji fonemów, jednak cecha ta



Rys. 2. Diagram czynności UML pokazujący proces opracowania systemu ARM; UML – Unified Modeling Language
Fig. 2. UML Activity Diagram showing the process of Automatic Speech Recognition system development; UML – Unified Modeling Language

nie powinna być uzyskiwana kosztem sztuczności mowy (wypowiedzi powinny być jak najbardziej naturalne, podobne do rzeczywistej mowy, która ma podlegać dekodowaniu podczas normalnego działania systemu).

Nagrania zawarte w zbiorze treningowym nie muszą odpowiadać pod względem leksykalnym docelowemu zakresowi słownictwa.

W początkowej koncepcji rozważano zebranie kilku godzin próbek głosu różnych osób poprzez nagranie ich wypowiedzi. Pozwoliłoby to uzyskać w nagraniach warunki akustyczne zbliżone do docelowych. Ze względu na ograniczenia czasowe oraz logistyczne przyjęta została inna koncepcja, zakładająca wykorzystanie fragmentów audiobooków. Ostatecznie przygotowany zbiór nagrań treningowych zawierał fragmenty książek czytane przez lektorów obu płci, wśród których było 10 kobiet i 10 mężczyzn. Należało je podzielić na pojedyncze pliki z krótkimi wypowiedziami. Podział na pliki został wykonany ręcznie. Łączna długość nagrań wyniosła nieco ponad 1 godzinę. Szczegóły dotyczące zawartości zbioru nagrań treningowych przedstawiono w Tabeli 1.

Zbiór testowy, w przeciwieństwie do zbioru treningowego, powinien składać się z nagrań zawierających wypowiedzi zbliżone do tych, które mają być rozpoznawane w konkretnym zastosowaniu.

Według zaleceń [20] zbiór testowy powinien stanowić ok. 10% zbioru treningowego, co przy nieco ponad godzinie nagrań treningowych oznaczało konieczność zgromadzenia kilku minut materiału. Przy tak niewielkich wymaganiach zdecydowano się zarejestrować przy użyciu mikrofonu wypowiedzi kilku mówców. W tym celu poproszono 6 osób (3 kobiety oraz 3 mężczyzn) w przedziale wiekowym 20–40 lat o odczytanie kilkudziesięciu komend z uprzednio przygotowanej listy.

Podczas badań do rejestrowania sygnału mowy użyto mikrofonu HAMA CS-461 [21]. Zbiór testowy został zarejestrowany dla odległości mówcy od mikrofonu wynoszącej ok. 0,5 m. Szczegóły dotyczące zawartości zbioru nagrań testowych zamieszczono w Tabeli 2.

2.4. Słownik fonetyczny

Słownik fonetyczny (.dic) to lista wszystkich unikalnych słów występujących w nagraniach, zarówno treningowych, jak i testowych wraz z ich transkrypcją fonetyczną.

Przygotowanie słownika fonetycznego obejmuje następujące kroki:

— utworzenie listy wszystkich unikalnych słów występujących w transkrypcjach zbioru treningowego i testowego,

Tabela 1. Szczegóły treningowego zbioru nagrań
Table 1. Details of the training database

głosy kobiece	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	razem
liczba plików	64	42	55	47	58	40	52	55	61	43	517
długość nagrań [min:sek]	3:50	3:31	3:35	2:54	3:03	3:07	3:02	3:16	3:05	3:07	30:30
głosy męskie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	razem
liczba plików	40	56	83	58	53	65	33	51	43	58	540
długość nagrań [min:sek]	4:07	3:48	3:54	4:08	3:33	2:58	2:53	3:31	2:57	3:01	34:50

– zdefiniowanie fonemów, które posłużą do opisu fonetyki poszczególnych słów,
– utworzenie reguł zamiany pisowni słów na odpowiednią wymowę, tj. sekwencje fonemów dla języka rozpoznawania mowy,
– utworzenie słownika fonetycznego.

Za pomocą programu *cmuclmtk* (CMU Sphinx) wygenerowano spis wszystkich unikalnych słów z transkrypcji zbioru treningowego i uzyskano w ten sposób 3897 wyrazów, natomiast z zestawu komend testowych otrzymano 54 kolejne słowa.

Należało ustalić zbiór wszystkich fonemów. Nie jest to sprawa oczywista, badacze przyjmują różne koncepcje nawet w obrębie jednego języka. Podczas przygotowywania niniejszej pracy wzięto pod uwagę dwie z nich: zaproponowaną w [15] oraz tzw. konwencję SAMPA [22]. W obu przypadkach wyróżnia się 37 fonemów dla języka polskiego, ale istnieją między nimi pewne różnice. Pierwsza propozycja przyjmuje istnienie fonemów ą i ę (zapisanych jako o~ i e~), zaś druga kwestionuje zasadność ich wyróżniania. Pojawiają się w niej z kolei dodatkowe fonemy *ki* oraz *gi* (np. w słowach *kiedy*, *zgiełk*). Pozostałe fonemy pokrywają się, włącznie z tzw. spółgłoską nosową tylnojęzykowo-miękkopodniebienną ŋ (np. w słowach *bank*, *sukienka*). Chcąc jak najdokładniej odzwierciedlić wymowę wyrazów ze słownika fonetycznego, celem możliwie najlepszego wytrenowania modelu akustycznego, zdecydowano się połączyć obie opisane powyżej koncepcje. Tym sposobem uzyskano zbiór 39 fonemów, które posłużyły do reprezentacji fonetycznej słów (tabela 3).

Każde unikalne słowo ze słownika zostało opisane przy pomocy pojedynczych fonemów z użyciem funkcjonalności translacji *g2p* (ang. *grapheme-to-phoneme*) pakietu *eSpeak* [23]. Uzyskany w ten sposób opis fonetyczny różnił się jednak od konwencji wymaganej przez CMU Sphinx. Pakiet *eSpeak* oparty jest na tzw. alfabecie Kirshenbauma [24], natomiast *sphinxtrain* oczekuje zapisu zbliżonego do Arpabet [25], w którym fonemy reprezentowane są tylko przy użyciu pojedynczych liter lub ich par rozdzielonych spacjami bez znaków interpunkcyjnych. Do konwersji fonemów z jednej konwencji na drugą przygotowano proste programy w języku C++.

Program *g2p eSpeak* ma zdefiniowany zestaw reguł (pl_rules) dla języka polskiego – w ramach pracy nie były one modyfikowane. Pojawiły się jednak problemy przy przetwarzaniu niektórych polskich znaków. Litery ś oraz ź zostały całkowicie pominięte, natomiast samogłoska ą odczytana została przez program błędnie. Ze względu na stosunkowo niewielką liczbę wyrazów w słowniku zawierających te litery zdecydowano się poprawić je ręcznie.

Należało jeszcze sprawdzić poprawność wygenerowanego słownika i skorygować go w razie potrzeby. Konieczne było to zwłaszcza w przypadku słów pochodzących z obcych języków, np. nazw własnych (*Missouri*, *Columbia*) występujących we fragmentach audiobooków. Niektórym słowom ze zbioru testowego dodano także alternatywną wymowę, wynikającą m.in. z kontekstu – np. jednoliterowe słowo *w* można wymawiać jako *f* lub *v* (*w prawo* [f p r a v o]/*w lewo* [v l e v o]).

2.5. Słownik wypełniaczy

W słowniku wypełniaczy (ang. *fillers*) znalazły się jedynie symbole oznaczające początek oraz koniec fragmentów ciszy.

2.6. Model statystyczny języka

Model statystyczny języka opisuje prawdopodobieństwa wystąpienia kombinacji różnych słów obok siebie w tekście referencyjnym reprezentującym język. Rozważono dwa warianty zbudowania modelu języka: utworzenie modelu n-gramowego albo gramatyki formalnej. Druga opcja może się wydawać trafnym wyborem do rozpoznawania komend z ograniczonego zbioru. Ze względu na łatwość przygotowania i planowany rozwój systemu ARM w stronę większej naturalności wypowiedzi w przyszłości oraz wyniki badań przeprowadzonych [10], zdecydowano się na n-gramowy model języka.

Pakiet *CMU Sphinx* zawiera program o nazwie *cmuclmtk* umożliwiający generowanie modeli n-gramowych.

Ponieważ duży nacisk kładziono na minimalizację błędu rozpoznawania pełnych komend, podjęto decyzję o narzuceniu pewnych ścisłych reguł syntaktycznych na sekwencje słów w komendach do rozpoznania. Komendy przyjmują następującą postać:

<komenda> = <czynność> + <obiekt> + <argumenty> (opcjonalnie).

W pracy użyto tekstu referencyjnego zawierającego docelowe komendy z uwzględnieniem wszystkich możliwych ich wariantów (tabela 4).

Wstawiając niektóre komendy wielokrotnie do tekstu referencyjnego, zadbano m.in. o to, by wyrównać prawdopodobieństwa wystąpienia wszystkich czasowników na pierwszej pozycji w wypowiedzi oraz zachować „symetrię" w przypadku par słów wyrażających kierunek ruchu, np. prawo/lewo, w górę/w dół (rys. 3).

W wygenerowanym modelu języka znalazły się 54 różne słowa (w tym symbole ciszy), 105 różnych bigramów i 166 trigramów. Każdemu z nich przyporządkowane zostały prawdopodobieństwa, reprezentowane w modelu również za pomocą liczb ujemnych – logarytmów dziesiętnych prawdopodobieństwa.

2.7. Konfiguracja treningu modelu akustycznego

Przed przystąpieniem do uczenia modelu akustycznego należy określić parametry konfiguracyjne programu *sphinxtrain*. Dokonuje się tego w pliku konfiguracyjnym *sphinx_train.cfg*. Trening opisywanego systemu przeprowadzono przy następujących wartościach parametrów (jeśli parametru nie wymieniono, miał on pozostawioną wartość domyślną):

```
$CFG_WAVFILE_SRATE = 16000.0;
$CFG_NUM_FILT = 25;
$CFG_LO_FILT = 130;
$CFG_HI_FILT = 6800;
$CFG_HMM_TYPE = '.cont.';
$CFG_FINAL_NUM_DENSITIES = 8;
$CFG_N_TIED_STATES = 200;
$CFG_LDA_MLLT = 'no';
$CFG_MMIE = 'no'.
```

Tabela 2. Szczegóły testowego zbioru nagrań
Table 2. Details of the test database

	K1	K2	K3	M1	M2	M3	razem
liczba plików	36	34	38	39	40	37	224
długość nagrań [min:sek]	1:05	0:56	0:58	1:27	1:17	1:05	6:48

Tabela 3. Konwencje zapisu fonemów i przykłady wymowy ze słownika
Table 3. Phoneme conventions and examples of pronunciation from dictionary

l.p.	fonem *SAMPA*	fonem *eSpeak*	fonem *Sphinx*	słowo	słowo *Sphinx*
1	i	i	i	zamknij	z a m k ni i j
2	I	y	y	prawy	p r a v y
3	e	E	e	lewe	l e v e
4	a	a	a	zgaś	z g a si
5	o	O	o	oko	o k o
6	u	u	u	zeruj	z e r u j
7	e~	E~	en	język	j en z y k
8	o~	O~	on	prawą	p r a v on
9	p	p, p;	p	pochyl	p o h y l
10	b	b, b;	b	obróć	o b r u ci
11	t	t, t;	t	wstręt	f s t r en t
12	d	d, d;	d	do	d o
13	k	k	k	wszystko	f sh y s t k o
14	k’	k;	ki	wielkie	v j e l ki e
15	g	g	g	głowę	g w o v e
16	g’	g;	gi	drugie	d r u gi e
17	f	f, f;	f	brew	b r e f
18	v	v, v;	v	włącz	v w on cz
19	s	s	s	stopni	s t o p ni i
20	z	z	z	zęby	z e m b y
21	S	S	sh	otwórz	o t f u sh
22	Z	Z	zh	obejrzyj	o b e j zh y j
23	s’	S;	si	głośnik	g w o si ni i k
24	z’	Z;	zi	źle	zi l e
25	x	x, C, h	h	ucho	u h o
26	ts	ts	ts	koniec	k o ni e ts
27	dz	dz	dz	bardzo	b a r dz o
28	tS	tS	cz	policzek	p o l i cz e k
29	dZ	dZ	dh	liczba	l i dh b a
30	ts’	ts;	ci	pięć	p j en ci
31	dz’	dz;	di	czterdzieści	cz t e r di e si ci
32	m	m, m;	m	zamknij	z a m k ni i j
33	n	n	n	piętnaście	p j e t n a si ci e
34	n’	n͡, n͡;	ni	ukłoń	u k w o ni
35	N	N	ng	dziękuję	di e ng k u j e
36	l	l	l	zapal	z a p a l
37	r	R	r	prawo	p r a v o
38	w	w	w	płacz	p w a cz
39	j	j	j	powiekę	p o v j e k e

```
\data\
ngram 1=54
ngram 2=105
ngram 3=166

\1-grams:                          \2-grams:                        \3-grams:
-0.7163 </s>     -2.9432           -0.0004 </s> <s> 0.0071          -0.3082 opuść ucho lewe
-0.7163 <s>      -2.9480           -1.2572 <s> bój 0.0071           -0.3082 opuść ucho prawe
-2.2199 brew     -1.5554           -1.2572 <s> głośnik 0.0209       -0.3082 otwórz powiekę lewą
-1.9732 bój      -1.7687           -1.2572 <s> obejrzyj 0.0071      -0.3082 otwórz powiekę prawą
-2.9732 czterdzieści -0.9026       -1.2572 <s> obróć 0.0209         -0.4008 piętnaście stopni do
-1.9732 do -1.8015                 -1.2572 <s> opuść 0.0071         -0.2247 piętnaście stopni w
-1.7646 głowę    -1.8898           -1.2572 <s> otwórz 0.0071        -0.0669 pięć stopni w
-1.9732 głośnik -1.7964            -1.2572 <s> pochyl 0.0140        -0.0212 pochyl głowę o
-2.5753 język    -1.1378           -1.2572 <s> podnieś 0.0071       -0.3118 pochyl oko lewe
-1.7428 lewe     -1.9110           -1.2572 <s> pokaż 0.0276         -0.3118 pochyl oko prawe
```

Rys. 3. Fragmenty modelu n-gramowego języka (n = 3)
Fig. 3. Excerpt from the n-gram model of language (n = 3)

3. Ewaluacja systemu

Ewaluacja systemu obejmowała dwa rodzaje badań: test systemu ARM będący integralną częścią procedury treningowej oraz badania z udziałem grupy testowej w warunkach rzeczywistych.

3.1. Test systemu ARM w ramach procedury treningowej

Test systemu ARM przewidziany w procedurze treningowej polega na dekodowaniu sygnału audio ze zbioru testowego za pomocą opracowanego systemu ARM, a następnie porównaniu rozpoznanego tekstu z transkrypcją. Na tej podstawie obliczane są wskaźniki jakości procesu rozpoznawania.

Jako wskaźniki jakości systemu przyjęto współczynniki błędu rozpoznawania słów WER (ang. *Word Error Rate*) oraz zdań SER (ang. *Sentence Error Rate*):

$$WER = \frac{S + D + I}{N_W} \tag{1}$$

$$SER = \frac{E}{N_S} \tag{2}$$

gdzie: S – liczba słów zastąpionych innymi (ang. *substitutions*), D – liczba słów pominiętych (ang. *deletions*), I – liczba słów niepotrzebnie wstawionych do zdania (ang. *insertions*), N_W – liczba słów występujących w zdaniu do rozpoznania, E – liczba błędnie rozpoznanych zdań, przy czym zdanie poprawnie rozpoznane oznacza, że nie było w nim ani jednego błędu typu zastąpienie, pominięcie lub wstawienie słowa, N_S – liczba zdań do rozpoznania.

W tabeli 5 przedstawiono wyniki testu w przypadku zastosowania dwóch wersji modelu języka LM, z których jedna była oparta na tekście referencyjnym nie w pełni zgodnym z nagraniami testowymi (LM1), a druga na tekście referencyjnym w pełni zgodnym z nagraniami (LM2).

Zbiór testowy składał się z 224 wypowiedzi, które łącznie zawierały 784 słowa. W wyniku pierwszego treningu uzyskano błąd rozpoznawania słów (WER) na poziomie 28,3% (222 błędy) oraz błąd rozpoznawania zdań (SER) wynoszący aż 51,8%, tj. 116 z 224 zdań zostało rozpoznanych niepoprawnie. Niezadowalające rezultaty wynikały z pewnej niezgodności w szyku słów w zdaniach zawartych w nagraniach testowych z ówczesną wersją modelu języka. Po naniesieniu odpowiednich poprawek udało się uzyskać znaczącą poprawę. Wartość WER spadła do 3,9%, co oznaczało błędne rozpoznanie zaledwie 31 z 784 słów przy jednoczesnym znacznym obniżeniu SER do 8,9% (20 z 224 zdań niepoprawnych).

Warto podkreślić, że uzyskane wartości WER i SER odnoszą się do konkretnego zbioru nagrań gromadzonego w laboratoryjnych warunkach akustycznych.

3.2. Badania w warunkach rzeczywistych

Głównym celem badań była ocena skuteczności rozpoznawania mowy przez system ARM w takich warunkach akustycznych, jakie mogą panować w ewentualnym docelowym zastosowaniu, w przypadku gdy mówcy stanowią grupę zróżnicowaną pod względem wieku oraz płci. Kolejnym celem badań była również ocena skuteczności rozpoznawania mowy przez system w przypadku, gdy mówca będzie znajdował się w różnych odległościach od mikrofonu.

Badania przeprowadzono zgodnie z diagramem czynności (rys. 4). Najpierw zadany tekst jest odczytywany przez mówcę, a zatem narządy mowy mówcy wytwarzają falę dźwiękową niosącą zadaną informację. Fala dźwiękowa z informacją rozchodzi się w otoczeniu testowym zgodnie z prawami propagacji fal i dociera do mikrofonu podłączonego do komputera, na którym działa opracowany system ARM. Oprócz fali dźwiękowej z zadaną informacją, do mikrofonu docierają jednocześnie fale dźwiękowe z innych źródeł, traktowane w tym badaniu jako zakłócenia. System ARM dekoduje dźwięk zarejestrowany przez mikrofon generując na wyjściu odpowiadający mu rozpoznany tekst. Następnie, rozpoznany tekst jest automatycznie porównywany z tekstem zadanym za pomocą odpowiedniego programu, a wynikiem tego porównania jest ocena, czy zdanie rozpoznano poprawnie czy błędnie. Ostatecznie, na podstawie całkowitej liczby błędnie rozpoznanych zdań oraz liczby wszystkich zdań obliczany jest automatycznie wskaźnik SER (równanie (2)).

Zadany tekst obejmował listę 76 prawidłowych poleceń dla robota (zdań) opisanych w punkcie 2.6. Ponieważ polecenia były odczytywane dwukrotnie, zadany tekst liczył 152 zdania. Badania przeprowadzono w pokoju o powierzchni 25 m^2 i wysokości 2,5 m. Podłoga nie była wyciszona wykładziną, a okna nie były w żaden sposób osłonięte podczas badania. Użyto mikrofonu konferencyjnego MXL AC-404 [26] położonego płasko na stole w odległości odpowiednio ok. 2 m i 1 m od ściany bocznej i tylnej (rys. 5).

W badaniu 1, realizującym pierwszy cel badawczy, wzięło udział pięcioro mówców:
— mówca nr 1 – mężczyzna, lat 22,
— mówca nr 2 – kobieta, lat 49,
— mówca nr 3 – mężczyzna, lat 28,
— mówca nr 4 – kobieta, lat 25,
— mówca nr 5 – mężczyzna, lat 53.

Tabela 4. Zestawienie komend do rozpoznawania
Table 4. Set of commands to be recognized

l.p.	czynność	obiekt	argument 1	argument 2	argument 3
1	obróć	głowę	–	o 15/45 stopni	w prawo/w lewo
2	obróć	brew	prawą/lewą	o 15/30 stopni	w prawo/w lewo
3	obróć	oko	prawe/lewe	o 15/30 stopni	w prawo/w lewo
4	pochyl	głowę	–	o 15 stopni	do przodu/do tyłu
5	pochyl	oko	prawe/lewe	o 15/30 stopni	w dół/w górę
6	przechyl	głowę	–	o 15 stopni	w prawo/w lewo
7	podnieś	ucho	prawe/lewe	–	–
8	opuść	ucho	prawe/lewe	–	–
9	otwórz	powiekę	prawą/lewą	–	–
10	zamknij	powiekę	prawą/lewą	–	–
11	pokaż	zaskoczenie	–	–	–
12	pokaż	wstręt	–	–	–
13	pokaż	zęby	–	–	–
14	pokaż	język	–	–	–
15	zapal	wyświetlacz	–	–	–
16	zapal	policzek	prawy/lewy	–	–
17	zgaś	wyświetlacz	–	–	–
18	zgaś	policzek	prawy/lewy	–	–
19	obejrzyj	się	–	–	–
20	ukłoń	się	–	–	–
21	uśmiechnij	się	–	–	–
22	bój	się	–	–	–
23	zezłość	się	–	–	–
24	płacz	–	–	–	–
25	włącz	głośnik	–	–	–
26	wycisz	głośnik	–	–	–
27	zeruj	wszystko	–	–	–
28	zeruj	*dowolny obiekt**	prawy/lewy**	–	–

*) komenda zeruj dotyczy obiektów: {głowę, brew, ucho, oko, powiekę, wyświetlacz, głośnik}
**) jeśli dotyczy

Tabela 5. Wyniki testu w ramach procedury sphinxtrain
Table 5. Test results obtained as a result of sphinxtrain procedure

	słowa			zdania		
	błędne	wszystkie	WER (%)	błędne	wszystkie	SER (%)
LM1	222	784	28,3	116	224	51,8
LM2	31	784	3,9	20	224	8,9

Tabela 6. Liczba błędnie rozpoznanych zdań E oraz SER dla badania 1
Table 6. The number of wrongly recognized sentences E and SER for the study 1

mówca	1 (M)	2 (K)	3 (M)	4 (K)	5 (M)	średnia	zakres
E	13	9	21	7	28	15,6	7–28
$\mathbf{N_s}$	152	152	152	152	152	152	–
SER [%]	8,5	5,9	13,8	4,6	18,4	10,3	4,6–18,4

Tabela 7. Liczba błędnie rozpoznanych zdań E oraz SER dla badania 2 (mówca 1)
Table 7. The number of wrongly recognized sentences E and SER for the study 2 (speaker 1)

odległość od mikrofonu	0,5 m	1,5 m	3 m
E	13	67	89
$\mathbf{N_s}$	152	152	152
SER [%]	8,5	44,1	58,6

W badaniu 2, dotyczącym drugiego celu badawczego, wziął udział tylko mówca nr 1, który mówił do mikrofonu z różnych odległości – 0,5 m, 1,5 m oraz 3,0 m.

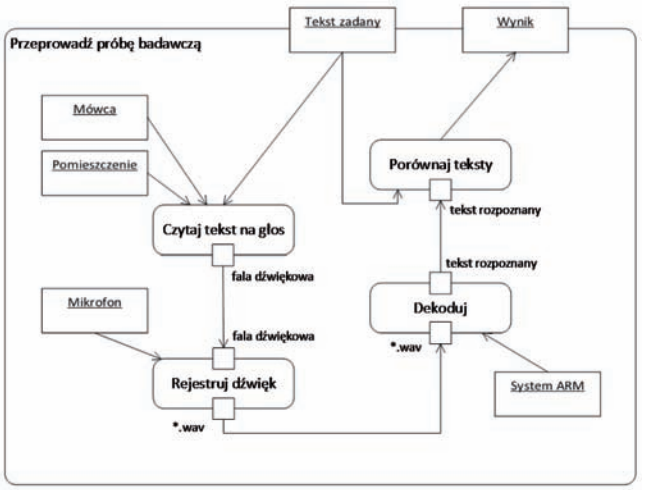
Wartości wskaźnika błędnie rozpoznanych zdań SER będące wynikiem badania 1 pokazano w tabeli 6, a wartości SER dla badania 2, w tabeli 7 oraz na wykresie (rys. 6).

Badania przeprowadzone zarówno w warunkach laboratoryjnych, jak i w warunkach rzeczywistych pokazały, że opracowany system ARM rozpoznaje mowę różnych osób, których głosy nie znajdują się w bazie treningowej modelu akustycznego. Błąd rozpoznawania pełnych zdań przyjmuje wartości od kilku do kilkunastu procent, a średnia dla badanej grupy użytkowników, w przypadku mówienia w niewielkiej odległości od mikrofonu, wynosi 10,3% (badanie 1). Warto w tym miejscu przypomnieć, że dla nagrań ze zbioru testowego osiągnięto współczynnik SER o zbliżonej wartości 8,9%. Nagrania ze zbioru testowego były jednak gromadzone w innych warunkach akustycznych i za pomocą mikrofonu kierunkowego, a następnie poddane selekcji, w trakcie której nagrania gorszej jakości zostały odrzucone.

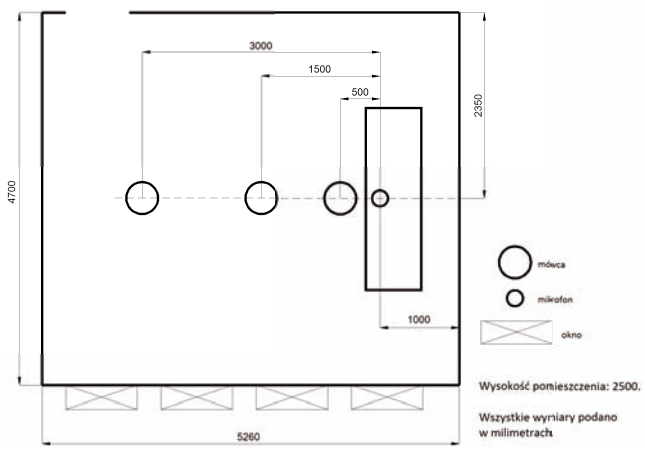
Badania potwierdziły przypuszczenie, że błąd rozpoznawania mowy rośnie wraz ze wzrostem odległości mówcy od mikrofonu. Użytkownik, który mówiąc bezpośrednio do mikrofonu uzyskał 8,5%, w przypadku testu z odległości 1,5 m został rozpoznany z SER na poziomie aż 44,1%. Przypuszczalnie gorszy rezultat otrzymany w przestronnym pomieszczeniu związany jest z występowaniem zjawiska pogłosu. Zwiększenie odległości do 3 m poskutkowało natomiast wzrostem błędu do 58,6%.

Należy zaznaczyć, że podane wartości błędu SER dla badań w warunkach rzeczywistych zostały otrzymane na małym zbiorze testowym, zatem obarczone są dużą niepewnością pomiaru. W celu potwierdzenia uzyskanych wyników konieczne jest przeprowadzenie badań na większym zbiorze testowym.

Podczas badań zaobserwowano także kilka najczęściej występujących błędów rozpoznawania. Niejednokrotnie wypowiedź *„zeruj oko prawe/lewe”* traktowana była przez system ARM jako *„zeruj ucho prawe/lewe”* ze względu na fonetyczne podobieństwo słów *„oko”* i *„ucho”*. Dość powszechnie powtarzającym się błędem było mylne rozpoznawanie słów *prawo/lewo* czy *prawą/lewą*. Pojawiał się on szczególnie często w przypadku wypowiedzi mówcy nr 5 i był główną przyczyną tak wysokiej wartości SER dla tego użytkownika. Kilkukrotnie wypowiedź *„zeruj głośnik”* dawała w rezultacie *„zeruj głowę”*. Zdarzało się również, zwłaszcza w dłuższych wypowiedziach, że tylko jedno ze słów było pomijane, co skutkowało zakwalifikowaniem tego zdania jako błędnie rozpoznanego.



Rys. 4. Diagram czynności UML pokazujący przebieg pojedynczej próby badawczej
Fig. 4. UML Activity Diagram showing flow of an individual trial during investigation

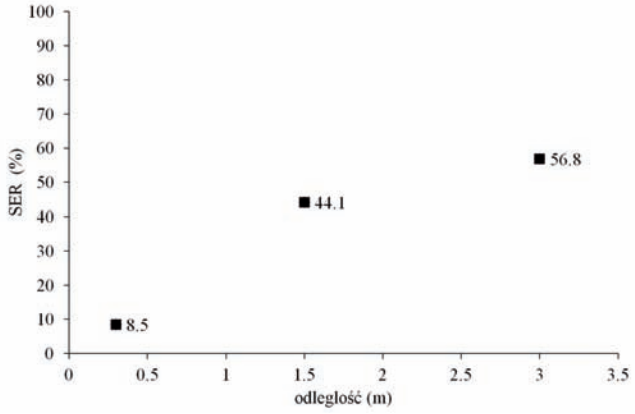


Rys. 5. Pomieszczenie testowe z badanymi położeniami mówcy względem mikrofonu
Fig. 5. Test room with investigated speaker positions relative to microphone

Interpretując uzyskane wyniki należy wziąć pod uwagę, że do wyszkolenia opracowanego systemu ARM użyto stosunkowo małej bazy treningowej nagrań. Czynnikiem, który także może mieć negatywny wpływ na skuteczność systemu jest przyjęty sposób opracowywania bazy treningowej, tj. przez wycinanie fragmentów audiobooków. Warunki akustyczne w rzeczywistości są znacznie gorsze niż w tych nagraniach.

4. Wnioski

W ramach pracy opracowano system automatycznego rozpoznawania mowy typu *command & control* dla języka polskiego przeznaczony dla robota społecznego.



Rys. 6. Zależność wskaźnika SER od odległości mówcy od mikrofonu (badanie 2)
Fig. 6. Sentence error rate (SER) dependency on speaker distance from microphone (study 2)

System bazuje na otwartym oprogramowaniu CMU Sphinx. Opracowano zbiór treningowy i testowy nagrań, zbiór głosek (fonemów) dla języka polskiego składający się z 39 fonemów, słownik fonetyczny, model 3-gramowy języka. Przeprowadzono trening modelu akustycznego.

Jakość opracowanego systemu określono poprzez wyznaczenie błędu rozpoznawania pełnych zdań SER podczas badań laboratoryjnych oraz badań w warunkach rzeczywistych. Podczas badań laboratoryjnych, przy materiale dźwiękowym pochodzącym od mieszanej grupy badawczej 6 osób w przedziale wiekowym 20–40 lat, uzyskano wartość SER na poziomie 9%. Podczas badań w środowisku rzeczywistym, przy mowie pochodzącej od grupy badawczej złożonej z 3 mężczyzn i 2 kobiet w wieku 20–50 lat oraz mówiących do mikrofonu z bliskiej odległości, uzyskano średnią wartość SER na poziomie 10%.

Przeprowadzono również badanie wpływu odległości mówcy od mikrofonu na uzyskiwaną wartość SER, która przy odległości 3 m wzrosła do blisko 60%.

W ramach dalszych prac planowane jest:
- rozszerzenie możliwości systemu, zarówno pod względem zasobu rozpoznawanego słownictwa oraz w kierunku umożliwienia użytkownikowi większej swobody wypowiedzi;
- zmniejszenie błędu rozpoznawania zdań w przypadku mowy odległego mówcy;
- przeprowadzenie pomiaru rozpoznawania zdań na większym niż dotychczas zbiorze testowym.

Podziękowania

Praca została wykonana w ramach projektu statutowego „Opracowanie prototypu głowy robota społecznego" w Przemysłowym Instytucie Automatyki i Pomiarów PIAP. Znaczący udział w opracowywaniu treningowego zbioru nagrań miała Pani Magdalena Dobrasiewicz, studentka Wydziału Elektroniki i Technik Informacyjnych Politechniki Warszawskiej.

Bibliografia

1. *Robotics 2020 – Multi-Annual Roadmap*. ICT 2016 (ICT 25 & ICT 26).
2. Fischinger D., Einramhof P., Papoutsakis K., Wohlkinger W., Mayer P., Panek P., Hofmann S., Koertner T., Weiss A., Argyros A., Vincze M., *Hobbit, a care robot supporting independent living at home: First prototype and lessons learned*. "Robotics and Autonomous Systems", Vol. 75, A, 2014, 60–78, DOI: 10.1016/j.robot.2014.09.029.
3. Gonzalez-Pacheco V., Malfaz M., Fernandez F., Salichs M.A., *Teaching human poses interactively to a social robot*. "Sensors", Vol. 13, No. 9/2013, 12406–12430, DOI: 10.3390/s130912406.
4. Nishimuta I., Yoshii K., Itoyama K., Okuno H.G., *Development of a robot quizmaster with auditory functions for speech-based multiparty interaction*. [in:] IEEE/SICE International Symposium on System Integration, SII 2014, 328–333, DOI: 10.1109/SII.2014.7028059.
5. Gomez R., Kawahara T., Nakamura K., Nakadai K., *Multiparty human-robot interaction with distant-talking speech recognition*. [in:] HRI'12 Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction. 439–446, 2012, DOI: 10.1145/2157689.2157835.
6. Gnjatović M., Tasevski J., Nikolić M., Mišković D., Borovac B., Delić V., *Adaptive multimodal interaction with industrial robot*. [in:] IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics, SISY 2012. 329–333, 2012, DOI: 10.1109/SISY.2012.6339538.
7. Rogowski A., *Analiza i synteza systemów sterowania głosowego w zautomatyzowanym wytwarzaniu*, Oficyna Wydawnicza Politechniki Warszawskiej, 2012.
8. Ondas S., Juhar J., Pleva M., Cizmar A., Holcer R., *Service robot SCORPIO with robust speech interface*. "International Journal of Advanced Robotic System", Vol. 10, No. 3, 2013, DOI: 10.5772/54934.
9. Jurafsky D., Martin J.H., *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, Upper Saddle River, N.J 2009.
10. Janicki A., Wawer D., *Automatic speech recognition for polish in a computer game interface*. [in:] 2011 Federated Conference on Computer Science and Information Systems (FedCSIS), 711–716, 2011.
11. Ziółko B., Jadczyk T., Skurzok D., Żelasko P., Gałka J., Pędzimąż T., Gawlik I., Pałka S., *SARMATA 2.0 Automatic Polish Language Speech Recognition System*, [in:] Sixteenth Annual Conference of the International Speech Communication Association, 2015.
12. Marasek K., Koržinek D., Brocki Ł., *System for Automatic Transcription of Sessions of the Polish Senate*. „Archives of Acoustics". Vol. 39, No. 4, 2014, 501–509, DOI: 10.2478/aoa-2014-0054.
13. Zygadło A., *System automatycznego rozpoznawania mowy polskiej na potrzeby robota społecznego*, 2016.
14. Gaida C., Lange P., Petrick R., Proba P., Malatawy A., Suendermann-Oeft D., *Comparing open-source speech recognition toolkits*. DHBW Stuttgart Technical Report, http://suendermann. com/su/pdf/oasis2014. pdf (2014).
15. Jassem W.: *Podstawy fonetyki akustycznej*. Państwowe Wydawnictwo Naukowe, Warszawa 1973.
16. [http://htk.eng.cam.ac.uk]
17. [http://cmusphinx.sourceforge.net] – CMU Sphinx, Project by Carnegie Mellon University
18. [https://sourceforge.net/projects/kaldi]
19. [http://cmusphinx.sourceforge.net/wiki/research] – Research Using CMUSphinx
20. [http://cmusphinx.sourceforge.net/wiki/tutorialam] – Training Acoustic Model For CMUSphinx
21. [https://pl.hama.com/000424610000/hama-mikrofon-stoj-cy-cs-461]
22. [http://www.phon.ucl.ac.uk/home/sampa/polish.htm]
23. [http://espeak.sourceforge.net] – eSpeak text to speech
24. [http://www.kirshenbaum.net/IPA/index.html] – Usenet IPA/ASCII transcription
25. [http://www.speech.cs.cmu.edu/cgi-bin/cmudict] – The CMU Pronouncing Dictionary
26. [http://www.mxlmics.com/microphones/web-conferencing/AC-404]

Automatic Speech Recognition System for Polish Dedicated for a Social Robot

Abstract: Automatic Speech Recognition system for Polish and dedicated for social robotics applications is presented. The system is based on free and open software library pocketsphinx (CMU Sphinx). Training and test databases were prepared with transcriptions; the training database comprised voices of 10 women and 10 men, and it was prepared based on audiobooks, whereas the test database comprised voices of 3 women and 3 men recorded in laboratory conditions as a part of the present work. A phoneme set for Polish consisting of 39 phonemes based on two popular sets from other researchers was prepared. The phonetic dictionary was obtained using grapheme-to-phoneme conversion from the eSpeak tool for speech synthesis. The language statistic model for the reference text including 76 commands was generated using cmuclmtk tool (CMU Sphinx). Training of the acoustic model and test of quality of speech recognition was conducted using the sphinxtrain tool (CMU Sphinx). The following error rates were obtained for laboratory conditions: 4% (WER) and 9% (SER). Next, investigations of the system in relevant real environment were conducted. The initial, tentative results are about 10% (SER) for the close distance of a speaker to a microphone, and about 60% (SER) for 3 m speaker-microphone distance. Directions of future works are formulated.

Keywords: automatic speech recognition, command and control, social robot

inż. Artur Zygadło
zygadlo.artur@gmail.com

Absolwent Wydziału Mechanicznego Energetyki i Lotnictwa Politechniki Warszawskiej (studia I stopnia na kierunku automatyka i robotyka, dyplom inż. 2016 r.). Obecnie student Wydziału Elektroniki i Technik Informacyjnych Politechniki Warszawskiej (studia II stopnia, kierunek Informatyka).

dr inż. Artur Janicki
A.Janicki@tele.pw.edu.pl

Adiunkt w Zakładzie Cyberbezpieczeństwa na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. Dyplom mgr. inż. elektroniki (1997 r.) i dr. inż. telekomunikacji (2004 r.). W 2014 r. odbył staż naukowy w ośrodku EURECOM w Sophia Antipolis, Francja. Autor lub współautor ponad 50 publikacji dotyczących przetwarzania sygnału mowy, w tym rozpoznawania mowy i mówcy. Promotor ponad 45 prac dyplomowych inżynierskich i magisterskich.

mgr inż. Przemysław Dąbek
pdabek@piap.pl

Absolwent studiów 1 stopnia na Coventry University, Wlk. Bryt., dyplom BEng w 2006 r., oraz studiów 1. i 2. stopnia w Centrum Kształcenia Międzynarodowego IFE Politechniki Łódzkiej, specjalność Mechanical Engineering and Applied Computer Science, dyplom mgr inż. w 2008 r. Od 2009 r. pracownik Przemysłowego Instytutu Automatyki i Pomiarów PIAP. Autor i współautor kilkunastu artykułów i referatów w dziedzinie dynamiki robotów mobilnych i pomiarów.